社会计算中的组织行为模式挖掘

苏　鹏◎著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

近年来，社会组织行为分析的研究主要集中在构建预测模型以预测组织可能的行为上。数据挖掘方法，特别是分类方法，近年来成为组织行为预测建模的主要方法。本书比较分析了主要的分类方法所建立的组织行为预测模型的性能，为不同情形下分类方法的恰当选择提供了依据。组织行为数据普遍存在类不平衡和误分类代价不一致的问题，这导致标准分类器所构建的组织行为预测模型性能较差。为此，在期望误分类代价这一指标下，本书研究了四种典型代价敏感学习方法基于不同标准分类器所构建的组织行为预测模型的性能，为不同情形下代价敏感学习方法的恰当选择提供了依据。另外，本书提出了一个新的适用于组织行为模式挖掘的代价敏感学习算法。最后，针对组织行为模式挖掘误分类代价易变且不易确定等特点，本书提出了基于代价曲线的个性化解决方案。

本书适合行为分析、数据挖掘、决策支持、商务智能等领域的学者、教师、研究生、本科生阅读使用，也可供承担管理社会组织职能的政府相关部门及事业单位的决策者与工作人员学习参考。

图书在版编目（CIP）数据

社会计算中的组织行为模式挖掘 / 苏鹏著. —北京：电子工业出版社，2019.1
ISBN 978-7-121-35264-5

Ⅰ. ①社… Ⅱ. ①苏… Ⅲ. ①计算机应用－社会科学－计算－组织管理－行为模式－研究 Ⅳ. ①C32

中国版本图书馆 CIP 数据核字（2018）第 240986 号

策划编辑：朱雨萌
责任编辑：朱雨萌　　特约编辑：丁福志
印　　刷：三河市兴达印务有限公司
装　　订：三河市兴达印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：720×1 000　1/16　印张：12　字数：211 千字
版　　次：2019 年 1 月第 1 版
印　　次：2019 年 1 月第 1 次印刷
定　　价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254750。

前 言

PREFACE

近年来，国际、国内各种社会组织的数量及活跃度快速增长，对各国的政治、经济等领域的影响日益加深。因此，迫切需要研究各种社会组织的行为模式与规律，为政府、企业等利益主体的科学决策提供坚实依据。随着互联网等新技术的飞速发展，信息变得空前丰富并易于存取，使得高效的计算技术，尤其是数据挖掘技术，在组织行为模式分析领域日益表现出巨大优势，并逐渐成为组织行为模式分析的主要方法。而将数据挖掘技术应用于组织行为模式分析也成为社会计算的一项重要研究内容。

目前，社会组织行为模式分析主要集中在构建预测模型以预测组织可能的行为上。数据挖掘方法，特别是分类方法，近年来成为组织行为预测建模的主要方法。本书比较分析了主要的分类方法所建立的组织行为预测模型的性能，为不同情形下分类方法的恰当选择提供了依据。组织行为数据普遍存在类不平衡和误分类代价不一致的问题，这导致标准分类器所构建的组织行为预测模型性能较差。为此，在期望误分类代价这一指标下，本书研究了四种典型代价敏感学习方法基于不同标准分类器所构建的组织行为预测模型的性能，为不同情形下代价敏感学习算法的恰当选择提供了依据。另外，本书提出了一个新的适用于本领域的代价敏感学习算法。实验结果表明，该算法优于其他五种常见的代价敏感学习算法。最后，针对本领域误分类代价易变且不易确定等特点，本书提出了一个基于代价曲线的个性化解决方案。该方案可使用户方便、直观地为给定数据集选择最优代价敏感学习算法——分类器组合。

尽管组织行为预测模型可提供相当准确的组织行为预测知识，但却不能提

供可被用户直接用来影响（抑制或鼓励）组织行为并因此获益的具体行动建议。这些行动建议又称为可操作知识，常常是用户切实需要的。可操作性是知识兴趣度的重要方面，使挖掘的模式具有可操作性是数据挖掘的中心主题之一。然而，尽管已有不少研究致力于其他类型的可操作知识发现，但是这种挖掘影响组织行为的可操作规则这一重要问题尚未被识别、定义和研究。为此，本书建立了一类新的组织行为模式挖掘问题——可操作行为规则挖掘。具体来说，本书提出了可操作行为规则挖掘问题的形式化定义，并提出多种可靠、有效的挖掘算法。值得强调的是，可操作行为规则挖掘技术在商务智能、企业管理等领域的应用前景也非常广阔。另外，本书还探讨了大数据背景下组织行为模式挖掘的研究框架。

作者

2018 年 7 月

目 录

CONTEN

第1章

概　述

1.1 社会计算的定义

随着信息数字化和网络化进程的不断加快，人们的行为轨迹越来越多地被记录下来，这使得利用计算技术观察和研究社会成为可能。

什么是社会计算？对于一个新兴的跨学科的研究领域往往仁者见仁、智者见智，很难给出一个公认的定义。一般而言，社会计算是指社会理论、信息系统、数学图论、计算网络、数据科学等交叉融合而成的一个研究领域，研究的是如何利用计算系统帮助人们进行沟通与协作，如何利用计算技术洞悉社会运行的规律与发展趋势。

1994 年，社会计算的概念第一次出现，Schuler 认为“社会计算可以是任何一种类型的计算应用，以软件作为媒介进行社交关系的应用”[1]。Dryer 等人将社会计算描述为一种理论概念，包括科学和技术方面，即“人类使用计算技术进行的社交行为和交互行为所产生的相互作用”[2]。Wang 等人认为社会计算是“信息技术和通信技术等促进了人类社会的研究和社会动态发展”。社会计算是指使用信息系统作为社会交互的场所，并使用信息系统作为数据收集和处理的空间，社会计算是在虚拟场所中的感知、交流和协作，社会计算需要把计算设备作为人与人之间交流的媒介，需要将人机交互设定成一个社会实践的环境，将理解社会过程作为交互系统的一部分工作[3]。

2009 年 2 月，美国哈佛大学的 David Laser 等 15 位美国学者在《科学》杂志联合发表了一篇具有里程碑意义的文章 *Computational Social Science*，该文章指出，“社会计算科学”这一研究领域正在兴起，人们将在前所未有的深度和广度上自动收集和利用数据，为社会科学的研究服务。之后，《科学》杂志相继发

表了多篇与社会计算相关的论文[4~7]，国外信息科学领域的多个重要学术期刊也出版了专刊，介绍与社会计算密切相关的社会媒体分析、社会学习、社会与经济计算等领域的发展情况[8~10]。在国内，社会计算的意义、理论、方法等也被不断深入探讨[11~14]。

社会计算利用计算技术研究传统意义上的社会学问题，使静态的人文知识动态化，使定性的讨论数字化，使孤立的知识网络化，最终使社会的发展和规划科学化。可以从两个角度来看待社会计算：一个是计算机或更广义的信息技术在社会活动中的应用，这一角度多限于技术层面，而且有很长的历史；另一个是社会知识，或者更具体的人文知识在计算机、信息技术中的使用和嵌入，反过来提高了社会活动的效益和水平。通过社会计算，将社会人文知识融入计算技术，用于分析和评估各种事关重大的社会发展政策和社会问题解决方案，开辟科学、技术、人文有机结合的一条新途径，是一项有价值的和长期性的研究。

1.2 社会计算研究的理论工具

社会计算研究所用的理论，主要有以下 3 类。

1. 从其他学科借鉴来的理论

（1）图论

从数学学科中引入的图论（Graph Theory）思想，为社会网络研究提供了持久的基础。

1736 年，29 岁的欧拉向圣彼得堡科学院递交了“哥尼斯堡的七座桥”的论文，在解答问题的同时，开创了数学的一个新分支——图论与几何拓扑，也由此开启了数学史上的新历程。19 世纪便已得出许多关于图论的重要结论，但直到 20 世纪 20 年代，图论才引起广大学者的注意并被广泛接受和传播。

图论即形象地用一些点及点与点之间的连线构成的图或网络来表示具体问题。利用图与网络的特点来解决系统中的问题，比用线性规划等其他模型来求解往往要简单、有效得多。图论就是研究图和网络模型特点、性质和方法的理论。图论在许多领域，如物理、化学、运筹学、计算机科学、信息论、控制论、网络理论、社会科学、经济管理等，都有广泛的应用，它已经广泛地应用于实际生活、生产和科学研究中。

（2）平衡论

平衡论（Balance Theory）是从社会心理学借鉴而来的理论。人们在寻找内心依靠的时候，或者说人们在寻找内心精神支柱的时候，曾经经历了唯物主义（物质至上）和唯心主义（精神至上）的实践失败。唯物主义主张物质第一、精

神第二。在这种情况下，如果人们过分追求物质，以自我为中心，就会内心膨胀，不能自拔，甚至产生暴力，丧失人性。没有束缚的物质追求会导致灾难。唯心主义主张精神第一、物质第二。过分追求唯心会使人产生消极的态度，相信命定论，那样人类就会失去最宝贵的能力——创造力。

单独的唯物或单独的唯心都不能达到内心的平衡，不能找到内心的精神支柱。只有根据自身的实际情况，在唯物主义和唯心主义中吸取精华，才能形成自己内心的平衡体系。唯物主义延伸即智慧、创造力，唯心主义延伸即心灵、（心存善意的）信仰，二者结合起来才能形成整体的平衡体系。

平衡论使人类可持续发展。人类只有精神和物质同步发展，才能达到可持续发展。也就是说，平衡论使人们认识到凡事都必须找到一个平衡点，这样才能使生活平稳和谐。平衡论的原则就是在法律和道德允许的基础上寻找平衡点，以达到自己内心的平和和生活的平稳。

（3）社会比较理论

社会比较理论（Social Comparison Theory）也是从社会心理学借鉴而来的理论。社会比较是一种普遍存在的大众心理现象。第一个系统地提出社会比较理论的人是费斯廷格。其理论的基本观点是：人人都自觉或不自觉地想要了解自己的地位如何、能力如何、水平如何。而一个人只有在社会中，通过与他人进行比较，才能真正认识自己和他人；只有“在社会的脉络中进行比较”，才能认识到自己的价值和能力，对自己做出正确的评价。社会比较能够使人清楚地了解自己和他人，找出自己和别人之间的差距，发现自己的长处和不足。由此可见，社会比较可以帮助人们认识自身，激发人们的行为动机。

心理学界对社会比较进行过很多研究，不少学者都提出了相关的理论。一般的观点认为，构成社会比较倾向应具备三个基本条件：一是人人具有想要清楚地评价自己的意义和能力的动机；二是如果有评价自己意义和能力的物理、客观的手段，则首先使用这种手段，如果找不到这种手段，则通过与他人进行比较来判断白己的意义和能力；三是与自己类似的人对自己的评价更有意义与

价值，所以容易被选作比较对象。

2. 本源的社会网络理论

（1）异质性理论

异质性理论（Heterophily Theory）包含弱连带优势、结构洞等概念。这一理论可以预见行动者在封闭的社会圈之外建立的连接如何帮助其获得多样化的知识及其他资源。

社会网络分析是连接微观和宏观层次的社会理论的主要工具，是通过对微观互动的分析来阐释其宏观含义的方法。格兰诺维特选取具有代表性的关于小规模互动的特定面向（人际连带强度）来阐释如何运用网络分析将这些面向关联到各种宏观现象上；进一步地，通过对社会网中“连带”（ties）的分析，提出“弱连带”在影响力与信息的传递、工作流动机会和社区组织等方面发挥着重要的作用。

结构洞是指社会网络中的空隙，即社会网络中的某个或某些个体和有些个体发生直接联系，但与其他个体不发生直接联系，即无直接联系或关系间断（disconnection）的现象，从网络整体看，好像网络结构中出现了“洞穴”。

个人在网络中的位置比关系的强弱更为重要，其在网络中的位置决定了个人的信息、资源与权力。因此，不管关系强弱，如果存在结构洞，那么将没有直接联系的两个行动者联系起来的第三者将拥有信息优势和控制优势，这样就能够为自己提供更多的服务和回报。因此，个人或组织要想在竞争中保持优势，必须建立广泛的联系，同时占据更多的结构洞，掌握更多的信息。

（2）结构角色理论

结构角色理论（Structural Role Theory）包含结构对等、结构内聚性、角色对等概念。这一理论可以对网络中的行动者如何相互影响对方的态度和行为等做出预见。

3. 大数据技术

大数据（Big Data）是指无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合。电子邮件、电子银行的支付记录、购物网站的消费记录、个人网页等互联网数据对研究人类和人类社会具有重要的价值。通过对这些数据进行计算和分析，原本不可捉摸的人类行为变得可被解析、描述和量化，甚至能够对其进行预测和控制。美国东北大学教授艾伯特·巴拉巴西经过十余年的数据分析发现，如果知道一个人过去的所有社会数据，那么，预测其未来行为的准确度将达到 93%。

大数据技术是指从各种各样类型的数据中快速获得有价值的信息的能力。

大数据技术包括：

（1）基础架构支持技术

基础架构支持技术主要包括支撑大数据处理的基础架构级数据中心管理、云计算平台、云存储设备及技术、网络技术、资源监控等技术。大数据处理需要拥有大规模物理资源的云数据中心和具备高效的调度管理功能的云计算平台的支撑。

（2）数据采集技术

数据采集技术是数据处理的必备条件。首先需要通过数据采集手段把信息收集上来，然后才能应用上层的数据处理技术。数据采集除各类传感设备等硬件和软件设施之外，主要涉及数据的 ETL（采集、转换、加载）过程，其能对数据进行清洗、过滤、校验、转换等各种预处理，将有效的数据转换成适合的格式和类型。同时，为了支持多源异构的数据采集、存储和访问，还须设计企业的数据总线，以方便企业各个应用和服务之间数据的交换与共享。

（3）数据存储技术

数据经过采集和转换之后，需要存储归档。针对海量的数据，一般可以采用分布式文件系统和分布式数据库的存储方式，把数据分布到多个存储节点，

同时还须提供备份、安全、访问接口及协议等机制。

（4）数据计算技术

与数据查询、统计、分析、预测、挖掘、图谱处理、BI 商业智能等相关的技术统称为数据计算技术。数据计算技术涵盖数据处理的方方面面，是大数据技术的核心。

（5）数据展现与交互技术

数据展现与交互技术在大数据技术中也至关重要，因为数据最终要被人们使用，为生产、运营、规划提供决策支持。选择恰当、生动、直观的展示方式能够帮助人们更好地理解数据及其内涵和关联关系，也能够更有效地解释和运用数据，发挥其价值。在展现方式上，除了传统的报表、图形之外，还可以结合现代化的可视化工具及人机交互手段，甚至最新的如 Google 眼镜等增强现实手段，来实现数据与现实的无缝对接。

1.3　社会计算的研究与应用领域

1. 社交网络服务

社交网络服务（Social Network Service，SNS）是根据哈佛大学心理学教授米尔格莱姆（Stanley Milgram）在 1967 年提出的六度分割理论（Six Degrees of Separation）建立的一种交流手段，即通过最多不超过六个人就可以认识一个陌生人。可以说，社会中人们之间存在一种不大明显的“弱关系”，使得相互之间都有极为紧密的联系。而社交网络服务就是将这一社会关系理论运用到网络上的一种创新，具体是指网络运营商在互联网上为用户建立沟通、交流与分享的社会性网络应用服务，也可代表社会已成熟普及的信息传播媒介载体。

Facebook（脸书）是美国的一个著名社交网络服务网站，创立于 2004 年 2 月 4 日。2018 年 3 月，Facebook 平均单日活跃用户为 14.5 亿人，比去年同期增长 13%；平均月度活跃用户人数为 22 亿人，比去年同期增长 13%。近年来，中国的社交网络服务网站也如雨后春笋般大量涌现，如人人网、开心网、赛我网、海内网等曾风靡一时，甚至成为大学生获得社会支持的一种重要途径。

社交网络服务的宗旨是为用户提供创造和维持人际交往关系的网络平台，概括地说，就是以交友为目的。而交友还可以分为熟人型和陌生型两类：熟人型即通过真实姓名及身份注册，将现实生活中已经存在的人际关系圈延伸至网络平台，利用六度分割理论，通过熟人找朋友或朋友的朋友，以不断扩大自己的交友范围；陌生型则根据用户的某种兴趣爱好或某个共同话题讨论热点等找到好友，通过共同话题进行沟通交流，进而结交某个领域的好友。可以说，熟人型和陌生型是相辅相成、相互融合的，并没有明确的区分。在现代社会，社

交网络是人们扩展人际交往的一个重要途径，值得深入分析研究。

2. 内容计算

除社交网络外，社会媒体也是分析、理解社会的重要素材，如新闻网站、论坛、博客、微博等。由于它们都以语言文字为主要展示形式，因此，从事内容计算（Content Computing）研究的学者需要掌握自然语言分析技术。目前，内容计算的热点包括舆情分析、基于内容的人际关系挖掘、微博应用等。

（1）舆情分析

传统地，对舆情的研究主要有两种方法：一是观察思辨；二是问卷调查。前者缺乏数据支持，后者采集的数据量亦有限。互联网技术为舆情分析提供了全新的技术路线，通过对各种社会媒体的跟踪与挖掘，结合传统的舆论分析理论，可以有效地观察社会的状态，并能辅助决策，及时发出预警。

（2）基于内容的人际关系挖掘

互联网中蕴含着大量公开的人名实体和人际关系信息。利用文本信息抽取技术可以自动抽取人名，识别重名，自动计算人物之间的关系，进而找出关系描述词，形成互联网世界的社会关系网。微软公司亚洲研究院的“人立方”就是一个典型的应用。

（3）微博应用

如果说人人网是中国的 Facebook，那么新浪微博则是中国的 Twitter（推特）。近年来，新浪微博发展迅猛，截至 2018 年 3 月，其月活跃用户数已增至 4.11 亿。在月活跃用户数突破 4 亿后，微博的用户增长迈上了一个全新的台阶。微博月活跃用户中来自移动端的比例高达 93%，日活跃用户则增至 1.84 亿。微博的用户使用时长也实现了增长。根据移动大数据服务商 QuestMobile 的最新数据，2018 年 3 月，微博用户的人均单日使用时长与去年同期相比增长了 21%。

微博同时具有社会网络和媒体平台的属性，它催生了信息生产和传播方式

的革命，对社会事件和人们的意识产生了很大的影响。微博明确地定位为平台，它提供开放的 API 接口，积极支持第三方应用的发展，基于微博的研究与开发已成为互联网学术界和产业界的热点。

3. 集体智慧

集体智慧（Collective Intelligence）简称“集智”，是一种共享的或群体的智能。在网络时代来临之前，集体智慧一直活跃在生物学、社会学、计算机科学、大众行为学等领域。随着 Web 2.0 的崛起和社会性软件的普及，集体智慧在社交网络服务、众包、分享、评论和推荐等领域也得到了广泛应用，典型案例包括维基百科、百度百科、百度知道、猪八戒网、任务中国、Threadless、InnoCentive、digg、iStockphoto、Mechanical Turk 等。这些互联网平台不仅帮助用户相互沟通、联系，更重要的是将用户组织起来，发挥他们的群体智慧，以协作的方式一起创造、加工和分享知识。越来越多的传统公司和组织开始使用各种集体智慧平台或工具，借助外部智慧解决复杂问题。

1.4 组织行为模式挖掘的研究内容

近年来，国际、国内各种社会组织的数量及活跃度快速增长，对各国的政治、经济等领域的影响日益加深。例如，石油输出国组织（OPEC）近年来频繁通过调整石油产量等行为干预国际石油价格，对世界经济产生了重大影响。因此，迫切需要研究各种社会组织的行为模式和规律，为政府、企业等利益主体的科学决策提供坚实的依据。

研究社会组织行为模式的传统方法是先通过亲自调查或调查问卷的方式收集数据，然后通过各种统计模型假设检验数据中的相关性。其主要不足是数据收集困难且分析手段单一。随着互联网等新技术的飞速发展，信息变得空前丰富并易于存取，这使得高效的计算技术，尤其是数据挖掘技术，在组织行为模式研究领域日益表现出巨大优势，并逐渐成为组织行为模式分析的主要方法。而将数据挖掘技术应用于组织行为模式分析也成为社会计算的一项重要研究内容。

1.4.1 组织行为预测

目前，组织行为模式挖掘的研究主要集中在构建预测模型以预测组织可能的行为。无论是一个国家的政府还是组织，准确预测其所关心的组织的行为对科学决策都具有重要意义。准确预测不仅可以使其有针对性地做好准备，而且可以使其预知可能采取的措施的后果，从而做出恰当取舍。例如，如果能对一些恐怖组织的暴力事件做出准确预测，政府就可以提前做好相应准备以显著减

少人民的生命财产损失，甚至可以防止暴力事件发生。

目前，预测建模的主要方法是统计机器学习与数据挖掘，如隐马尔可夫（HMM）方法、概率逻辑程序（PLPs）法和以 CONVEX 算法[15]为代表的标准分类方法。较早的研究工作[16~18]采用隐马尔可夫模型预测组织行为，这种方法尽管可以可靠地预测行为结果，但却难以推断模型中的哪些因素导致了被预测的结果。另外，模型的构建依赖很长时间段内的事件收集，因而费时费力。为发现特定组织在一个可能“世界”模型中最可能的行为或行为集合，文献[17]基于行为规则和概率逻辑程序（PLPs）提出了两个算法，即 SemiHOP 和 SemiHOP_binary 算法。该方法具有很好的可解释性，但没有基于任何指标进行模型的性能分析。

为了准确、迅速地预测组织行为，文献[15]提出了 CONVEX 算法。CONVEX 算法把每个样本看作一个矢量对，其包含环境矢量和行为矢量。环境矢量包含组织相关的环境变量值，而行为矢量包含组织的行为变量值。为预测行为矢量，CONVEX 算法使用矩阵空间中的距离函数来计算给定环境矢量和其他环境矢量的相似度。因此，CONVEX 算法实际上可看作 k-最近邻算法的一个变体。基于社会文化基准数据集 MAROB（Minorities At Risk Organizational Behavior）[19]，CONVEX 算法取得了超过 95%的预测准确率。

虽然以 CONVEX 算法为代表的分类方法在组织行为预测准确率方面取得了不错的效果，但组织行为数据中普遍存在的类不平衡和非一致误分类代价问题会严重阻碍标准分类器所建立的预测模型的性能，而代价敏感学习方法是解决该问题的有效手段。

近年来，代价敏感学习一直是机器学习和数据挖掘领域的研究热点。如果考虑代价模型的抽象粒度，代价敏感学习方法可分为两类：样本依赖代价敏感学习方法（如文献[20~25]）和类依赖代价敏感学习方法（如文献[26~30]）。前者假定每个样本错分到特定类具有不同的代价，后者假定每类样本的错分代价都相同。在大部分应用中，为每一类别确定误分类代价比为

每个样本确定误分类代价要容易得多。考虑代价信息在学习过程中的应用方式，代价敏感学习方法也可以分为两类：第一类应用贝叶斯风险理论指定待分类样本到具有最低期望风险的类（如文献[30]）；第二类在构造分类器之前，通过采样[31]或给样本赋权[28]改变类分布。第二类方法更通用，因为它不必输出类概率估计，而且不必重构分类器。考虑吸收代价因素对学习过程的改变程度，代价敏感学习方法也可以分为直接方法（如文献[27, 28]）和间接方法（如文献[31]）。

某些代价敏感学习方法，如采样[31]、调整决策阈值[30]、MetaCost[27]和样本加权法[28]等在实际中得到了广泛应用。作为处理类不平衡问题的常见方法，上采样通过产生一定数量的少数类样本同时保持多数类样本数目不变来实现训练集类分布的平衡。该方法的主要缺点是容易导致过拟合[31]。同样作为处理类不平衡问题的常见方法，下采样则通过删除一定数量的多数类样本同时保持少数类样本数目不变来实现训练集类分布的平衡。下采样一般用作算法比较的基准算法[31]。MetaCost 通过对任意分类器包装一个最小化代价使其实现代价敏感，是一种非常有效且具有良好可伸缩性的代价敏感学习方法[27]。

调整决策阈值（Threshold-moving）方法（如文献[26, 30]）的主要思想是向高误分类代价类移动决策阈值，以使具有高误分类代价的样本更容易被识别。使用调整后的阈值，待分类样本会被指定为产生最小期望误分类代价的类。调整决策阈值方法是最直接、简单的代价敏感学习方法，因为它不改变分类器的学习过程。与其他代价敏感学习方法相比，当误分类代价改变时，其分类模型无须做任何改变。调整决策阈值方法在解决类不平衡问题时和采样方法一样有效[32]。样本加权（Instance-weighting）法[28]的基本思想是，在保持训练集样本总权重不变的情况下，根据每类的误分类代价对样本加权。样本加权法与先前的基于改变类先验的方法相比一样有效，而实现更加简单。

尽管提出了很多代价敏感学习方法，但是研究代价敏感学习基本理论的文献却不多。对于分类属性有两个取值的分类问题，文献[29]证明了一个理

论，即为了使用一个从标准的非代价敏感的学习算法获得的分类器做出最优的代价敏感分类决策，该如何改变训练集负样本的比例。文献[26]建立了样本分布、类先验概率、每类误分类代价和决策阈值间的联系。这一联系表明了上述常用方法具有理论上的等价性，但是文献[33]表明它们之间的精确关系是复杂的，并因任务和方法而异。由于本领域数据集的特点与其他领域明显不同，如样本稀少、不平衡程度高和属性多为离散等，导致不能把在其他领域应用代价敏感学习算法的经验直接运用到本领域。因此，通过实验研究全面比较评价上述方法在组织行为预测建模领域的性能表现，可为在本领域有效应用代价敏感学习方法并研发适合本领域的高性能代价敏感学习算法提供经验依据。

1.4.2　可操作行为规则挖掘

组织行为预测模型产生如“如果满足条件 c，则组织 g 将会做出行为 b”的预测知识，但却不能直接明确地向用户建议可影响（抑制或鼓励）组织行为并因此获益的具体行动，而这种关于行动建议的知识常常是用户切实需要的。

考虑一个安全信息学中的例子。在一个假设场景中，某组织频繁发动暴力恐怖袭击，致使某政府面临的国际、国内压力剧增。在这种情况下，政府当然希望通过采取某些有效行动来抑制恐怖行为。由此，类似下述的以规则形式呈现的可靠建议或预测将因具有重要价值而引起政府的巨大兴趣：

“如果政府将针对某组织的致命武力打击的程度从级别 1（不使用）提高到级别 2（阶段性使用），将与某组织达成协议的程度从级别 1（谈判）提高到级别 3（重大让步），则某组织针对国内目标的恐怖袭击的烈度将从级别 3 以 70%的概率下降到级别 2，或以 20%的概率下降到级别 1，或以 10%的概率保持不变；针对国际目标的恐怖袭击的烈度将从级别 3 以 30%的概率下降到级别 2 或

以 20%的概率下降到级别 1 或以 50%的概率保持不变。”

这种规则向用户提供明确的行动建议以影响组织行为并因此获益。此类知识称作可操作知识，即人们可据其行动，或可导致某种行动，或可使某些事情发生的知识[34]。可操作性是知识兴趣度的重要方面，需要基于用户的主观评价量化，以促进决策制定[35]。让挖掘出的模式具有可操作性是数据挖掘的中心主题之一[34]。

目前，不少研究工作已致力于发现不同类型的可操作知识（如文献[36~38]）。可操作知识一般以可操作规则为形式。如果用户可基于某条规则采取可使其获益的行动，则该条规则被认为是可操作的[39]。

例如，为提高银行客户的收益，文献[36]提出了从特定分类规则对中构建行动规则（Action rule）的方法。首先，属性被分为两类：稳定属性和灵活属性。前者（后者）包含其值不能（能）被银行改变或影响的属性。其次，分类规则以灵活属性优先的方式从决策表中抽取。最后，一类新的规则即行动规则从该分类规则库中构建。一个灵活属性值的改变表示采取了一个行动。一条行动规则定义为 $[(\omega)\wedge(\alpha\rightarrow\beta)]\Rightarrow(\gamma\rightarrow\varphi)$，其中，$\omega$ 是稳定属性值的合取，$(\alpha\rightarrow\beta)$ 表示灵活属性值的改变（行动），$(\gamma\rightarrow\varphi)$ 是该行为的预期后果。行动规则指明某些属性需要如何改变，以使不受欢迎的对象转到受欢迎的类别中。文献[40~45]对行动规则进行了进一步研究。

属性值的改变将招致代价。文献[46]提出了行动规则的代价和可行性的概念，并给出了一个基于搜索图的以最低代价构建可行行动规则的方法。文献[47]将最低代价行动规则定义为有趣的行动规则。文献[48]提出了一个构建有趣行动规则的启发式算法。文献[49]提出了一个产生有趣行动规则的方法，该方法将一个启发式策略应用于抽取行动规则的行动森林算法。

尽管选择的分类算法不同，以上的行动规则挖掘方法均基于一对特定的分类规则或一条分类规则产生一条可操作规则。这一策略的主要缺点是可能遗漏一些有趣的可操作规则。为克服这一缺点，文献[50]提出了一个在支持度—置

信度—代价框架下直接从数据库中挖掘行动规则的策略。文献[51]提出了一个挖掘关联类型行动规则的方法。文献[52]提出了一个采用自底向上策略而无须使用预先存在的分类规则的行动规则挖掘方法。

为帮助某公司设计一个可提高其收益的直销计划，文献[37]提出了一个使用“角色模型”产生建议和计划的即时方法。角色模型是一个典型的可产生用户建议的案例库。对每个寻求建议的新客户，一个最近邻算法被用来发现具有高成本效益及实现可能性的计划，从而将该客户转到受欢迎的角色模型中。该方法并不事先准备规则，因而当产生行动建议时将付出很高的计算代价。

为发现可操作知识以促进客户关系管理，文献[38, 53]提出了将一个客户从不受欢迎的状态重分类到受欢迎状态的方法。该方法后处理决策树以最大化期望净收益。然而，此类方法有可能漏掉一些高净收益的行动建议。为解决这一问题，文献[38]提出了构建多棵决策树的策略，其中，每棵树包含“硬”属性的不同子集。但是，为获得最优行动建议，决策树的数量可能会非常大，当“硬”属性很多时尤其如此。

另外，文献[54]从系统的决策制定的角度提出了一个可操作知识发现（Actionable Knowledge Discovery，AKD）的形式化视图，并对提出的四类相应的一般 AKD 框架进行了形式化与论证。

上述可操作规则挖掘工作所提出的方法都不能用于挖掘影响组织行为的可操作规则。第一，以往方法所处理的数据集的元组表示所关心实体的成员对象（如某公司的客户），而组织行为数据集的元组表示对所关心组织的不同时间段的观察。第二，以往方法仅能处理单个二值决策属性，而本领域须考虑多个多值决策属性。第三，以往方法所建议行动的直接后果是将所关心实体的部分成员对象从非期望决策类重分类到期望决策类（如把银行客户从“非忠诚”类重分类到“忠诚”类），而本领域需要的行动建议的直接后果是转变所关心组织的行为状态（无特定目标状态），以使用户获得满意收益。总之，

挖掘影响组织行为的可操作规则（可操作组织行为规则）这一重要问题尚未被识别、定义和研究。

另一类相关工作是关联分类（Associative Classification，AC），其集成了关联规则挖掘与分类两类数据挖掘任务，以构建预测模型（分类器）。AC 算法一般能获得很大的规则集，而其中很多规则是冗余和有误导性的[55, 56]。多种剪枝方法被用来降低 AC 分类器的规模，其中，规则排序方法被大多数 AC 分类器使用。一个决定规则先后次序的重要参数是规则前件的长度。一些文献（如文献[56, 58, 59]）所提出的 AC 算法倾向于一般规则（前件较短的规则），这导致了较低的分类准确率。相反，一些文献（如文献[60, 61]）所提出的 AC 算法倾向于具体规则（前件较长的规则），这会减少误分类机会。对生成规则集的优化同样也是可操作规则挖掘所面临的关键问题。

一条 AC 规则可表示为 $[\alpha] \Rightarrow (\gamma)$，其中，$\alpha$ 是特征属性值，γ 是决策属性的一个值。与可操作规则挖掘相比，AC 和其他传统的基于规则的分类并不考虑属性值的改变。因此，传统分类方法不管在问题定义还是形式体系上都不能提供关于如何改变属性值以使用户获得满意收益的可操作建议。

为此，本书建立了一类新的组织行为模式挖掘问题——可操作（组织）行为规则挖掘，这具有重要的学术价值。具体来说，本书提出了可操作行为规则挖掘问题的形式化定义，并提出了两个有效、可靠的可操作行为规则挖掘算法。值得强调的是，可操作行为规则挖掘也可应用于国家、群体等其他实体的行为建模。

在本书提出的框架中，可操作行为规则可表示为 $[(\alpha \to \beta)] \Rightarrow [(\gamma \to \varphi, p)]$，其中，$p$ 表示行动的相应效果的概率。形式上，可操作行为规则与以往的可操作规则相比有两大重要不同。第一，前者的 α 与 γ 是当前观察的属性值。第二，前者的后件可涉及多个多值行为属性，而后者的后件仅涉及单个二值决策属性。因此，在形式体系和具体方法方面，可操作行为规则挖掘与其他的可操作规则挖掘都是极其不同的。表 1-1 比较了不同方法的异同。

表 1-1　可操作行为规则挖掘与以往方法的比较

项　　目	可操作行为规则挖掘	其他可操作规则挖掘	基于规则的分类
数据集中的对象含义	对某实体的观察	某实体的成员对象	对某实体的观察
决策属性数目	多个	单个	单个
决策属性可能值的数目	多个	两个	多个
是否需要最小置信度阈值	否	是	是
是否需要最小支持度阈值	是	是	是
是否需要指定行动的效用	是	是	否
规则形式/输出	$[(\alpha \to \beta)] \Rightarrow [(\gamma \to \varphi, p)]$	$[(\omega) \wedge (\alpha \to \beta)] \Rightarrow (\gamma \to \varphi)$	$[\alpha] \Rightarrow (\gamma)$
建议行动的期望直接影响	改变实体的行为状态	重分类实体的某些成员对象	无

可操作行为规则挖掘在国家安全、公共政策及商务智能等多个领域将有广泛的应用和发展前景。

1. 可操作行为规则挖掘将有助于我国政府制定有效的国家安全策略

随着经济全球化和国家关系的多边化、国际组织的多样化，国家安全问题日益超出其原有的内涵，向国家和社会生活的各个领域扩展，甚至成为关乎建立国际政治经济新秩序的重大战略问题。可操作行为规则挖掘可为我国政府建议有效措施，以影响所关心国家、组织的行为，从而维护国家根本利益。例如，其可为我国政府建议有效措施以达到以下效果：①影响相关大国的行为从而使其与我国的摩擦保持低强度和可控性；②抑制甚至制止一些组织的分裂行为；③抑制甚至制止一些恐怖组织的各种恐怖行为。

2. 可操作行为规则挖掘将有助于我国各级政府制定完善的公共政策

我国正步入公共需求快速增长和深刻变化的重要时期，利益主体和社会结构正在发生重要改变，社会矛盾和社会问题日益突出，并已成为世界上收入差距比较大，城乡差距比较严重，就业、公共医疗、义务教育、社会保障等公共需求和公共服务方面问题比较突出的国家之一。可操作行为规则挖掘可为我国各级政府提供各种有效政策建议，以影响目标群体的行为，从而获得良好的社会效益。例如，其可为某地政府建议有效措施，以抑制该地的犯罪行为，从而

维护社会安定。

3. 可操作行为规则挖掘将为商务智能开辟新应用并推动其进一步发展

世界经济的快速发展、商务活动的日益频繁，以及数据挖掘、数据仓库等新技术的不断涌现，对商务的智能化提出了越来越高的要求。可操作行为规则挖掘作为一种新的行为建模技术将为商务智能开辟新应用。例如，在客户关系管理（CRM）领域，可为某公司建议有效措施，以促进客户群体（作为整体）的购买行为。

1.5　本书的结构与内容

本书对组织行为模式挖掘技术与方法进行了深入研究。从整体上，本书内容可分为两部分：第一部分是组织行为预测建模，主要围绕本领域普遍存在的类不平衡与非一致误分类代价问题，以提高预测模型质量为目的展开研究；第二部分建立了一类新的组织行为模式挖掘问题——可操作行为规则挖掘。各章的具体内容如下：

第 1 章介绍了社会计算的定义、研究理论工具、研究与应用领域，并提出了组织行为模式挖掘的概念与研究内容。

第 2 章研究了组织行为预测建模。首先，介绍了基于相似度的组织行为预测建模方法——CONVEX 算法。其次，比较分析了主要的分类方法，及其所建立的组织行为预测模型的性能。再次，为解决本领域普遍存在的类不平衡与非一致误分类代价问题，研究了四种典型代价敏感学习方法基于不同标准分类器建立的预测模型在不同情形下的性能。另外，基于上采样方法提出了一个适用于本领域的对不同正样本采用不同复制策略的高性能代价敏感算法。最后，提出了一种基于代价曲线的针对本领域类不平衡与非一致误分类代价问题的有效个性化解决方案。

第 3 章建立了一类新的组织行为模式挖掘问题——可操作行为规则挖掘。首先，提出了可操作行为规则挖掘问题的形式化定义。其次，提出两个可操作行为规则挖掘算法 MABR-1 和 MABR-2。最后，提出了可操作行为规则挖掘算法（模型）的经验验证方法并基于该方法验证了提出算法的有效性。

第 4 章致力于探讨建立精确的可操作行为规则挖掘的计算模型，设计有效、高效的规则挖掘算法。具体来说，为消解规则的冲突，提出了一种新的规则排序方法；为精确描述行为样本对可操作行为规则的非一致支持强度，提出了一个规则支持度的样本加权模型，以及一个相应的挖掘算法；为直接处理数值行为属性，提出了直接基于数值行为属性的可操作行为规则挖掘的新定义，以及一个相应的规则挖掘算法；为充分利用先验知识及显著减少挖掘算法的时间复杂度，提出了一个基于贝叶斯网络的可操作行为规则挖掘方法及相应算法；为显著减少挖掘算法的时间复杂度，还提出了一个基于决策树的近似规则挖掘算法。

第 5 章致力于探讨大数据背景下的组织行为模式挖掘。具体讨论了面临的挑战、应对策略及实现方案等。

第 6 章总结了本书作者的研究工作、研究成果及创新点。

第 2 章

组织行为预测

2.1 基于相似度的组织行为预测方法

2.2 基于分类的组织行为预测方法

2.3 代价敏感组织行为预测建模

近年来，机器学习与数据挖掘方法（如文献[62~70]），尤其是各种分类方法（如文献[71~86]）成为构建组织行为预测模型的主要方法。2.1 节介绍了有代表性的基于相似度的组织行为预测方法（CONVEX 算法）。2.2 节比较分析了主要的分类方法所建立的组织行为预测模型的性能。2.3 节为解决本领域普遍存在的类不平衡与非一致误分类代价问题，研究了四种典型代价敏感学习方法基于不同标准分类器建立的组织行为预测模型在不同情形下的性能。另外，基于上采样方法提出了一个适用于本领域的对不同正样本采用不同复制策略的高性能代价敏感算法。最后，提出了一种基于代价曲线的针对本领域中类不平衡与非一致误分类代价问题的有效个性化解决方案。

2.1 基于相似度的组织行为预测方法

2.1.1 组织行为的矢量模型

假定有一个包含某组织多年数据的数据集，每个过去的行为可由一对矢量描述。其中，环境矢量包含组织相关的环境变量值（包括其他组织采取的对该组织有影响的行动信息），行动矢量包含组织相关的反映组织所采取行动的行动变量值。

假定某用户想识别该组织在当前或假定环境下可能采取何种行动，查询矢量用来描述组织所处的或假定所处的环境，则用户必定会对查询矢量相关的行动矢量感兴趣。例如，相关行动矢量可能告诉用户组织会实施炸弹袭击而不会实施绑架。

形式化地，假定存在一个属性集合 A，每个属性 A_i 的值域为 $\mathrm{dom}(A_i)$。任

一组织 g 有一相关环境模式 $\mathrm{CS}(g)=(C_1,\cdots,C_i,\cdots,C_m)$，行动模式 $\mathrm{AS}(g)=(A_i,\cdots,A_j,\cdots,A_n)$，其中，任 $C_i, A_j \in A, \{C_1,\cdots,C_m\}\bigcap\{A_1,\cdots,A_n\}=\varnothing$。

例如，在 MAROB 数据集中，有一个 284 个属性组成的环境模式，其包含域为{是，否}的属性 FORSTFINSUP（国外财政支持）、域为{是，否}的属性 DEMORG（组织民主）等。环境变量又分为几个类别：组织采用暴力的程度、组织参与政治进程的程度、组织遭受歧视的程度和组织拥有的资源类型。数据集只包含类别属性而不包括数值属性。

一个 g−行为是一个环境矢量−行动矢量对：$<(c_1,\cdots,c_m),(a_1,\cdots,a_n)>$，其中，$c_i \in \mathrm{dom}(C_i)$，$a_i \in \mathrm{dom}(A_i)$。组织 g 的过去行为 PB(g)是 g−行为的有限集。

表 2-1 是某 MAROB 组织的一个 g−行为的例子。LEAD 表示组织领导类型，ORGPOP 表示组织在所在国家的受欢迎程度，ARMATTACK 表示组织是否参与武装袭击，HOSTAGE 表示组织是否参与绑架。

表 2-1　某 MAROB 组织的行为

环境属性				行动属性		
年份	LEAD	ORGPOP	DEMORG	FORSTFINSUP	ARMATTACK	HOSTAGE
1993	4	2	1	0	1	0
1994	3	2	1	0	1	0
1995	3	2	1	0	0	0
1996	3	2	1	0	1	0
1997	3	2	1	0	0	0

关于 g 的查询矢量 $\boldsymbol{q}$ 可表示为 $(c_1,\cdots,c_m)$。给定 PB(g) 和 $\boldsymbol{q}$，可以从该组织的过去行为中学习得到合适的行动矢量 $(a_1,\cdots,a_n)$。

2.1.2　CONVEX 算法

CONVEX$^{\mathrm{kNN}}$ 和 CONVEXMerge 算法使用矩阵空间的距离函数预测行动矢量。

1. 距离函数

每个环境矢量或查询矢量都对应 m 维矢量空间 $\mathrm{dom}(C_1)\times\cdots\times\mathrm{dom}(C_m)$ 中的一个点。假定函数 d 表示这一矢量空间中的距离度量，则 d 仅须满足一般距离函数须满足的 3 个公理：

$$d(\boldsymbol{x},\boldsymbol{y})=0$$

$$d(\boldsymbol{x},\boldsymbol{y})=d(\boldsymbol{y},\boldsymbol{x})$$

$$d(\boldsymbol{x},\boldsymbol{z})\leqslant d(\boldsymbol{x},\boldsymbol{y})+d(\boldsymbol{y},\boldsymbol{z})$$

其中，$\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}$ 是环境矢量。

假定 $(c_1,\cdots,c_m),(c_1',\cdots,c_m')$ 是矢量空间 $\mathrm{dom}(C_1)\times\cdots\times\mathrm{dom}(C_m)$ 的成员，可以选择以下 6 种常见的距离函数之一作为 d。

（1）欧几里得距离（Euclidean distance）：

$$d_{\mathrm{EUC}}((c_1,\cdots,c_m),(c_1',\cdots,c_m'))=\sqrt{(c_1-c_1')^2+\cdots+(c_m-c_m')^2}$$

（2）兰氏距离（Canberra distance）：

$$d_{\mathrm{CAN}}((c_1,\cdots,c_m),(c_1',\cdots,c_m'))=\sum_{i=1}^{m}|c_i-c_i'|/\left(|c|+|c'|\right)$$

若 $|c|+|c'|=0$，则 $d_{\mathrm{CAN}}((c_1,\cdots,c_m),(c_1',\cdots,c_m'))=0$。

（3）切比雪夫距离（Chebyshev distance）：

$$d_{\mathrm{CHEB}}((c_1,\cdots,c_m),(c_1',\cdots,c_m'))=\max_i(|c_i-c_i'|)$$

（4）余弦距离（Cosine distance）：

$$d_{\mathrm{COS}}((c_1,\cdots,c_m),(c_1',\cdots,c_m'))=(\sum_{i=1}^{m}|c_i\times c_i'|)/(|c|\times|c'|)$$

（5）海明距离（Hamming distance）：

$$d_{\mathrm{HAM}}((c_1,\cdots,c_m),(c_1',\cdots,c_m'))=\sum_{i=1}^{m}\mathrm{isDiff}(c_i,c_i')$$

其中，若 $c_i = c_i'$，则 $\text{isDiff}(c_i, c_i') = 0$，否则 $\text{isDiff}(c_i, c_i') = 1$。

（6）曼哈顿距离（Manhattan distance）：

$$d_{\text{MAN}}((c_1, \cdots, c_m), (c_1', \cdots, c_m')) = \sum_{i=1}^{m} | c_i - c_i' |$$

2. CONVEX^{kNN} 算法

CONVEX^{kNN} 算法首先根据 d 从 PB 中找到与 $\boldsymbol{q}$ 距离最近的 $k(k \geqslant 1)$ 个环境矢量。对任一行动 a_j，找到 k 个最临近环境矢量相应的行动矢量的 a_j 值 $v_{j_1}, \cdots, v_{j_k}$。用 v 表示 v_{j_i} 的均值，如果 v 为整数，则返回 v，否则返回区间 $[[v],[v]]$。

以下内容描述了 CONVEX^{kNN} 算法。Insert 函数用于将若干元素插入列表中，并保持列表中元素按升序排列。

输入：g, PB(g), q, d, A_j, List：长度为 k 的列表，其成员被初始化为+∞

输出：A_j 的预测值或预测区间

```
1. for each (cv, av) ∈ PB(g)
2.     dist←d(q, cv)
3.     If dist＜d(q, List[k])
4.         Insert((cv, av), List)
5. v←average of List[1].av[j],…, List[k].av[j]
6. if v is an integer
7.     return v
8. else
9.     return [⌊v⌋,⌈v⌉]
```

例如，假定查询矢量是（4,1,1,0），距离测度函数采用欧几里得距离，k 取 1，则与查询矢量距离最近的是 1993 年的数据。显然，CONVEX^{kNN} 会预测 ARMATTACK=1，HOSTAGE=0。换句话说，该组织将会明确地使用武装袭击

而不是绑架作为策略。这并不意味着绑架事件不会发生，而是该组织不会将其作为斗争策略。

现在假定查询矢量是（3, 2, 1, 1），距离测度函数采用欧几里得距离，k 取 4，则 1994—1997 年的数据是最近邻居。CONVEX^{4NN} 会预测 HOSTAGE=0，ARMATTACK=[0,1]，这表明 ARMATTACK 属性的取值存在不确定性，要么取 0，要么取 1。

类似地，假定 k 取 3，则与查询矢量距离最近的是 1994—1997 年的数据。然而，CONVEX^{3NN} 仅会选择前三条数据，并基于这三条数据的行动矢量作出和上例相同的预测。

下面将证明 CONVEX^{kNN} 算法的时间复杂度与 g 的过去行为，也即 PB(g) 的大小存在线性关系。因为 k 一般取值很小，所以 CONVEX^{kNN} 运行很快。

命题 2-1 假定 d 的计算时间为常数，则算法 CONVEX^{kNN} 的时间复杂度为 $O(k \times \mathrm{PB}(g))$。

证明 算法的循环至多运行 PB(g)次，每次循环迭代需要调用一次时间复杂度为 $O(k)$ 的 Insert 函数。故得证。

3. CONVEXMerge 算法

与 CONVEX^{kNN} 算法一样，CONVEXMerge 算法的基本思想也是根据 d 从 PB 中找到与 $\boldsymbol{q}$ 距离最近的 $k(k \geqslant 1)$ 个环境矢量。然而，这些邻居的重要性和其与 $\boldsymbol{q}$ 的距离成反比。

考虑 $\boldsymbol{q}$ 在 PB(g)中的两个最近的邻居，其中一个邻居与 $\boldsymbol{q}$ 的距离为 1，另一个邻居与 $\boldsymbol{q}$ 的距离为 10。在这种情形下，根据最近邻居赋予行动属性 A_i 的值的优先级必定比根据第二近邻居赋予的高。CONVEXMerge 就是根据这一思想得来的。

假定 $k \geqslant 1$，$\boldsymbol{q}$ 的 k 个最近邻居 kNN($\boldsymbol{q}$,PB(g)) 是 g–行为 $(c_1, a_1), \cdots, (c_k, a_k)$。

若 $\sum_{(c_i,a)\in \mathrm{kNN}(q,PB(g))} d(c_i,q)=0$，也即所有 k 个最近邻居与 q 的距离为 0，则 A_i=a 的概率为

$$P(A_i=a|\boldsymbol{q},\mathrm{PB}(g))=\left|\left\{(c_i,a)\middle|(c_i,a)\in \mathrm{kNN}(\boldsymbol{q},\mathrm{PB}(g))\right\}\right|/k$$

否则，若 k 个距离不全为 0，则分不同情况定义 A_i=a 的概率：

① 若 $\left\{(c_i,a)\middle|(c_i,a)\in \mathrm{kNN}(\boldsymbol{q},\mathrm{PB}(g))\right\}=\mathrm{kNN}(\boldsymbol{q},\mathrm{PB}(g))$（所有最近邻居的 A_i 值均为 a），则

$$P(A_i=a\,|\,\boldsymbol{q},\mathrm{PB}(g))=1$$

② 若 $\left\{(c_i,a)\,|\,(c_i,a)\in \mathrm{kNN}(\boldsymbol{q},\mathrm{PB}(g))\right\}=\varnothing$（没有任何最近邻居的 A_i 值为 a），则

$$P(A_i=a\,|\,\boldsymbol{q},\mathrm{PB}(g))=0$$

③ 在所有其他情况下，有

$$P(A_i=a\,|\,\boldsymbol{q},\mathrm{PB}(g))=1-\sum_{(c_i,a)\in \mathrm{kNN}(q,PB(g))} d(c_i,\boldsymbol{q})\Big/\sum_{i=1}^{k} d(c_i,\boldsymbol{q})$$

$\sum_{(c_i,a)\in \mathrm{kNN}(q,PB(g))} d(c_i,q)\Big/\sum_{i=1}^{k} d(c_i,q)$ 的分母是 $\boldsymbol{q}$ 与所有 k 个最近邻居的距离和。分子是 $\boldsymbol{q}$ 与所有其 A_i=a 的最近邻居的距离和。因此，分子越小，其 A_i=a 的最近邻居与和 $\boldsymbol{q}$ 最相似的 g–行为越“近”（或相似）。因为小距离标志高概率，所以用 1 减去该比值。

下面举一个简单的例子。假定 $\boldsymbol{q}$ 为（0,0），有三个环境矢量，$c_1=(0,1)$，$c_2=(0,2)$，$c_3=(1,1)$。假定 A_i 取 0 或 1，且对 c_1 取 0，对 c_2 与 c_3 取 1，则有 $d_{\mathrm{EUC}}(\boldsymbol{q},c_1)=1$，$d_{\mathrm{EUC}}(\boldsymbol{q},c_2)=2$，$d_{\mathrm{EUC}}(\boldsymbol{q},c_3)=\sqrt{2}$。假定 k=3，则有

$$P(A_i=0\,|\,\boldsymbol{q},\mathrm{PB}(g))=1-(1+2)/(1+2+\sqrt{2})=0.32$$

$$P(A_i=1\,|\,\boldsymbol{q},\mathrm{PB}(g))=1-(1+\sqrt{2})/(1+2+\sqrt{2})=0.68$$

直觉上，A_i=1 的概率比 A_i=0 的概率高，因为有两个与 $\boldsymbol{q}$ 距离分别为 1 和

2 最近邻居有 A_i=1，第三个与 $\boldsymbol{q}$ 距离为 $\sqrt{2}$ 的最近邻居以概率 $P(A_i=1)$ 有 $A_i=1$。这导致了 $A_i=0$ 的概率减小。

对任一 A_i，CONVEXMerge 算法都会为任一 $a \in \mathrm{dom}(A_i)$ 计算 A_i=a 的概率，返回最高概率值或概率分布。以下内容展示了最终算法。

输入：g, PB(g), q, d, A_j, List: 长度为 k 的列表，其成员被初始化为+∞

输出：$\{(a, P(A_i = a \mid q, PB(g)))a \in \mathrm{dom}(A_i)\}$

```
1. for each (cv, av) ∈ PB(g)
2.     dist← d(q, cv)
3.     if dist＜d(q, List[k])
4.         Insert((cv, av), List)
5. return {(a, P(A_i = a | q, PB(g))) | a ∈ dom(A_i)}
```

CONVEXMerge 算法仅比 CONVEX^{kNN} 慢一点，其时间复杂度包含一个附加的乘法因子 dom(A_j)。这是因为必须计算每个 $\mathrm{dom}(A_j)=a$ 的概率。

命题 2-2 假定 d 的计算时间为常数，则算法 CONVEXMerge^{k} 的时间复杂度为 $O(k \times \mathrm{PB}(g) \times |\mathrm{dom}(A_j)|)$。

证明 略。

2.2　基于分类的组织行为预测方法

基于分类的组织行为预测模型的质量和所选择的分类方法密切相关，因此，有必要对各种分类方法所构建的组织行为预测模型的性能进行比较与评价，以作为实际应用中分类方法的选择依据。

2.2.1　分类方法

分类是数据挖掘中一个重要的分支，有着广泛的应用，如医学疾病判别、垃圾邮件过滤、垃圾短信拦截、客户分析等。

分类问题可以分为归类和预测两类。

归类：指对离散数据的分类，例如，根据一个人的笔迹判断这个人是男还是女，这里的类别只有两个（男，女）。

预测：指对连续数据的分类，例如，预测明天 8 点的空气湿度，空气湿度是一个连续值，它不属于某个有限集合。预测也称回归分析，在金融等领域有着广泛的应用。

分类过程有两个步骤：构造模型，利用训练数据集训练分类器；利用建好的分类器模型对测试数据进行分类。

本节选择朴素贝叶斯（Naive Bayes，NB）、支持矢量机（Support Vector Machine，SVM）、人工神经网络（Artificial Neural Network，ANN）、k−最近邻（k-Nearest Neighbor，kNN）分类、决策树（Decision Tree，DT）、随机森林（Random

Forest，RF）与关联分类 7 种主要的分类方法构建组织行为预测模型。

1. 朴素贝叶斯

朴素贝叶斯是基于贝叶斯理论的经典的概率分类器。对给定类别，若特征间不相互依赖，则朴素贝叶斯可看作最优分类器。尽管实现简单，但朴素贝叶斯在很多应用中有优异表现。

（1）朴素贝叶斯分类的原理

对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。在没有其他可用信息时选择条件概率最大的类别，就是朴素贝叶斯的思想基础。

根据上述分析，朴素贝叶斯分类的流程如图 2-1 所示。

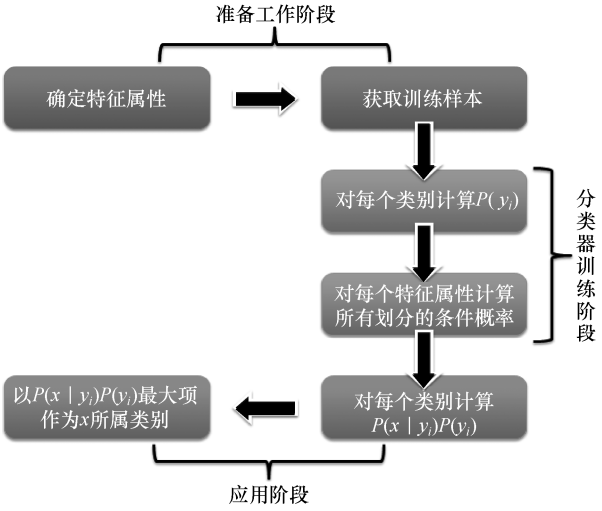


图 2-1　朴素贝叶斯分类的流程

由图 2-1 可以看出。整个朴素贝叶斯分类分为 3 个阶段：

第一阶段——准备工作阶段。这个阶段的任务是为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。

这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量在很大程度上由特征属性、特征属性划分及训练样本质量决定。

第二阶段——分类器训练阶段。这个阶段的任务是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并记录结果。其输入是特征属性和训练样本，输出是分类器。这一阶段是机械性阶段，根据前面讨论的公式可以由程序自动计算完成。

第三阶段——应用阶段。这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。

（2）估计类别下特征属性划分的条件概率

计算各个划分的条件概率 $P(a|y)$是朴素贝叶斯分类的关键步骤，当特征属性为离散值时，只要很方便地统计训练样本中各个划分在每个类别中出现的频率即可用来估计 $P(a|y)$。

当特征属性为连续值时，通常假定其值服从高斯分布（也称正态分布）。因此，只要计算出训练样本中各个类别中此特征项划分的各均值和标准差，代入上述公式即可得到需要的估计值。

当 $P(a|y)=0$ 时，也就是当某个类别下某个特征项划分没有出现时，分类器质量将大大降低。为了解决这个问题，引入拉普拉斯校准，它的思想非常简单，就是对每个类别下所有划分的计数加 1，当训练样本集数量充分大时，不会对结果产生影响，并且解决了上述频率为 0 的问题。

2. 支持矢量机

支持矢量机方法[70, 88]是 20 世纪 90 年代初由 Vapnik 等人根据统计学习理论提出的一种新的机器学习方法，它以结构风险最小化原则为理论基础，通过适

当地选择函数子集及该子集中的判别函数，使学习机器的实际风险达到最小，保证了通过有限训练样本得到的小误差分类器，对独立测试集的测试误差仍然较小。支持矢量机在解决小样本、非线性及高维模式识别中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中。本书为支持矢量机选择 RBF 核函数以简化模型构建，并使用网格搜索和留一交叉验证来发现最优超参数 C 和 γ。

（1）支持矢量机的基本思想

首先，在线性可分情况下，在原空间寻找两类样本的最优分类超平面。在线性不可分的情况下，加入松弛变量进行分析，通过使用非线性映射将低维输入空间的样本映射到高维属性空间，使其变为线性情况，从而使得在高维属性空间采用线性算法对样本的非线性进行分析成为可能，并在该特征空间中寻找最优分类超平面。其次，通过使用结构风险最小化原理在属性空间构建最优分类超平面，使得分类器达到全局最优，并在整个样本空间的期望风险以某个概率满足一定上界。

支持矢量机突出的优点表现在：①基于统计学习理论中结构风险最小化原则[在保证分类精度（经验风险）的同时，降低学习机器的 VC 维（Vapnik-Chervonenkis Dimension），可以使学习机器在整个样本集上的期望风险得到控制]和 VC 维理论是由统计学理论定义的有关函数集学习性能的一个重要指标，其目的是研究学习过程一致收敛的速度和推广性，具有良好的泛化能力，即由有限的训练样本得到的小的误差能够保证使独立的测试集仍保持小的误差；②支持矢量机的求解问题对应的是一个凸优化问题，因此局部最优解一定是全局最优解；③核函数的成功应用，将非线性问题转化为线性问题求解；④分类间隔的最大化，使得支持矢量机算法具有较好的鲁棒性。

（2）最优分类面和广义最优分类面

支持矢量机是从线性可分情况下的最优分类面发展而来的，基本思想可用图 2-2 来说明。对于一维空间中的点、二维空间中的直线、三维空间中的平面，

以及高维空间中的超平面，图中实心点和空心点代表两类样本，H 为它们之间的分类超平面，H_1，H_2 分别为过各类中离分类面最近的样本且平行于分类面的超平面，它们之间的距离 Δ 称为分类间隔（Margin）。

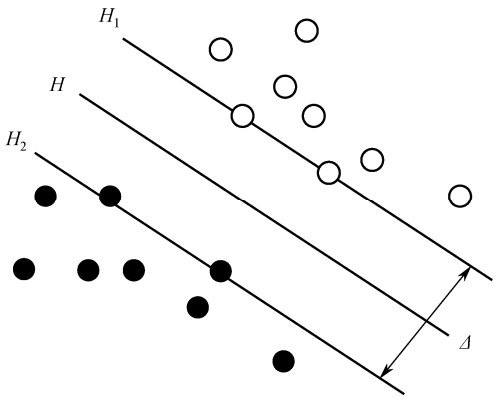


图 2-2　最优分类面

最优分类面要求分类面不但能将两类正确地分开，而且使分类间隔最大。将两类正确地分开是为了保证训练错误率为 0，也就是经验风险最小（为 0）。使分类空隙最大实际上就是使推广性的界中的置信范围最小，从而使真实风险最小。推广到高维空间，最优分类线就成为最优分类面。

设线性可分样本集为 $(x_i, y_{i\_}), i=1,\cdots,n, x\in\mathbf{R}^d, y\in\{+1,-1\}$ 是类别符号。d 维空间中线性判别函数的一般形式为类别符号。d 维空间中线性判别函数的一般形式为 $g(\boldsymbol{x})=\boldsymbol{w}\cdot\boldsymbol{x}+b$（$\boldsymbol{w}$ 代表 Hilbert 空间中的权矢量；b 代表阈值），分类线方程为 $\boldsymbol{w}\cdot\boldsymbol{x}+b=0$。将判别函数进行归一化，使两类所有样本都满足 $|g(\boldsymbol{x})|=1$，也就是使离分类面最近的样本的 $|g(\boldsymbol{x})|=1$，此时分类间隔等于 $2/\|\boldsymbol{w}\|$，因此使间隔最大等价于使 $\|\boldsymbol{w}\|$（或 $\|\boldsymbol{w}\|^2$）最小。要求分类线对所有样本正确分类，就是要求它满足

$$y_i[(\boldsymbol{w}\cdot\boldsymbol{x})+b]-1\geqslant 0, i=1,2,\cdots,n \tag{2-1}$$

满足式（2-1），并且使 $\|\boldsymbol{w}\|^2$ 最小的分类面称为最优分类面，过两类样本中离分类面最近的点且平行于最优分类面的超平面 H_1，H_2 上的训练样本点就称

为支持矢量（Support vector），因为它们“支持”了最优分类面（见图 2-3）。

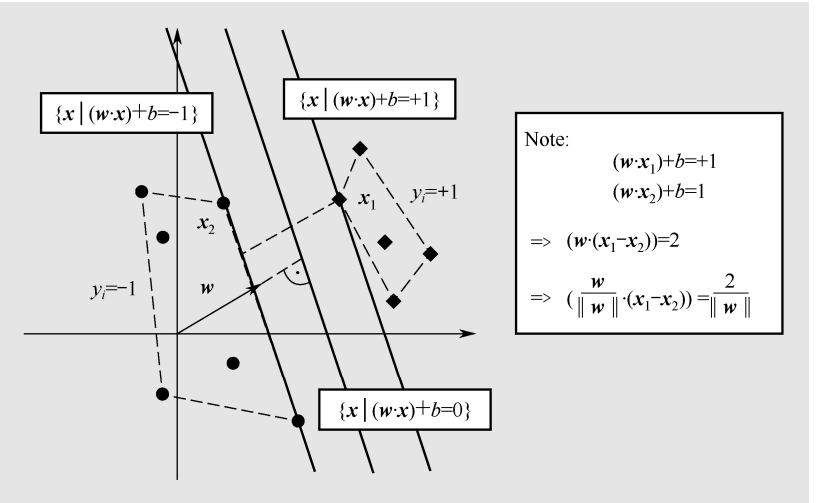


图 2-3　支持矢量

利用拉格朗日优化方法可以把上述最优分类面问题转化为如下较简单的对偶问题，即在约束条件

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{2-2a}$$

$$\alpha_i \geqslant 0, i = 1, 2, \cdots, n \tag{2-2b}$$

下面对 α_i（对偶变量即拉格朗日乘子）求解下列函数的最大值：

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i x_j) \tag{2-3}$$

若 α^* 为最优解，则

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha^* y \alpha_i \tag{2-4}$$

即最优分类面的权系数矢量是训练样本矢量的线性组合。

这是一个不等式约束下的二次函数极值问题，存在唯一解。根据 kühn-

Tucker 条件，解中将只有一部分（通常是很少一部分）α_i 不为零，这些不为 0 解所对应的样本就是支持矢量。求解上述问题后得到的最优分类函数是

$$f(x)=\operatorname{sgn}\{(\boldsymbol{w}^*\cdot\boldsymbol{x})+b^*\}=\operatorname{sgn}\{\sum_{i=1}^{n}\alpha_i^* y_i(x_i\cdot\boldsymbol{x})+b^*\} \tag{2-5}$$

根据前面的分析，非支持矢量对应的 α_i 均为 0，因此式（2-5）中的求和实际上只对支持矢量进行。b^* 是分类阈值，可以由任意一个支持矢量通过式（2-1）求得（只有支持矢量才满足其中的等号条件），或通过两类中任意一对支持矢量取中值求得。

从前面的分析可以看出，最优分类面是在线性可分的前提下讨论的，在线性不可分的情况下，就是某些训练样本不能满足式（2-1）的条件，因此可以在条件中增加一个松弛项参数 $\varepsilon_i \geqslant 0$，变成

$$y_i[(\boldsymbol{w}\cdot x_i)+b]-1+\varepsilon_i \geqslant 0,\ i=1,2,\cdots,n \tag{2-6}$$

对于足够小的 $s>0$，只要使

$$F_\sigma(\varepsilon)=\sum_{i=1}^{n}\varepsilon_i^\sigma \tag{2-7}$$

最小，就可以使错分样本数最小。对应线性可分情况下使分类间隔最大，在线性不可分情况下可引入约束：

$$\|\boldsymbol{w}\|^2 \leqslant c_k \tag{2-8}$$

在约束条件式（2-6）、式（2-8）下对式（2-7）求极小值，就得到了线性不可分情况下的最优分类面，称为广义最优分类面。为方便计算，取 $s=1$。

为使计算进一步简化，广义最优分类面问题可以进一步演化成在式（2-6）的约束条件下求下列函数的极小值：

$$\phi(\boldsymbol{w},\varepsilon)=\frac{1}{2}(\boldsymbol{w},\boldsymbol{w})+C(\sum_{i=1}^{n}\varepsilon_i) \tag{2-9}$$

其中，C 为某个指定的常数，它实际上起控制错分样本惩罚程度的作用，实现

错分样本的比例与算法复杂度之间的折中。

（3）支持矢量机算法的非线性映射

对于非线性问题，可以通过非线性交换转化为某个高维空间中的线性问题，在变换空间求最优分类超平面。这种变换可能比较复杂，因此这种思路在一般情况下不易实现。然而在上面的对偶问题中，不论是寻优目标函数[式（2-3）]还是分类函数[式（2-5）]，都只涉及训练样本之间的内积运算（$x \cdot x_i$）。设有非线性映射 $\Phi : \mathbf{R}^d \to \boldsymbol{H}$ 将输入空间的样本映射到高维（可能是无穷维）的特征空间 $\boldsymbol{H}$ 中，当在特征空间 $\boldsymbol{H}$ 中构造最优超平面时，训练算法仅使用空间中的点积，即 $\phi(x_i) \cdot \phi(x_j)$，而没有出现单独的 $\phi(x_i)$。因此，如果能够找到一个函数 K 使得

$$K(x_i \cdot x_j) = \phi(x_i) \cdot \phi(x_j)$$

则在高维空间实际上只须进行内积运算，而这种内积运算是可以用原空间中的函数实现的，甚至没有必要知道变换中的形式。根据泛函的有关理论，只要一种核函数 $K(x_i \cdot x_j)$ 满足 Mercer 条件，它就对应某一变换空间中的内积。因此，在最优超平面中采用适当的内积函数 $K(x_i \cdot x_j)$ 就可以实现某一非线性变换后的线性分类，而计算复杂度却没有增加。此时目标函数变为

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j)$$

相应的分类函数变为

$$f(x) = \operatorname{sgn}\{\sum_{i=1}^{n} \alpha_i^* y_i K(x_i \cdot x_j) + b^*\}$$

算法的其他条件不变，这就是支持矢量机。

概括地说，支持矢量机就是通过某种事先选择的非线性映射将输入矢量映射到一个高维特征空间，在这个特征空间中构造最优分类超平面。在形式上，支持矢量机分类函数类似于一个神经网络，输出是中间节点的线性组合，每个

中间节点对应一个支持矢量，如图 2-4 所示。

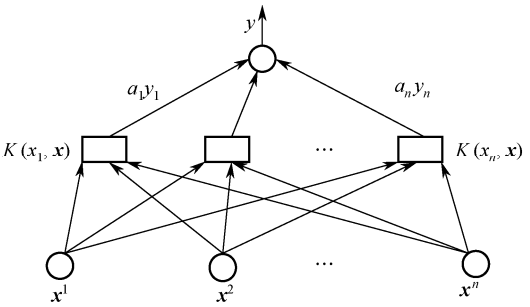


图 2-4　支持矢量机

其中，输出（决策规则）：$y=\mathrm{sgn}\{\sum_{i=1}^{n}\alpha_i y_i K(\boldsymbol{x}\cdot x_i)+b\}$，权值 $w_i=\alpha_i y_i$，$K(\boldsymbol{x}\cdot x_i)$ 为基于 s 个支持矢量 $x_1,x_2,\cdots,x_s$ 的非线性变换（内积），$\boldsymbol{x}=(x^1,x^2,\cdots,x^d)$ 为输入矢量。

（4）核函数

选择满足 Mercer 条件的不同内积核函数，就构造了不同的支持矢量机，这样也就形成了不同的算法。目前研究最多的核函数主要有 3 类：

① 多项式核函数：

$$K(\boldsymbol{x},x_i)=[(\boldsymbol{x}\cdot x_i)+1]^q$$

其中，q 是多项式的阶次，所得到的是 q 阶多项式分类器。

② 径向基函数（RBF）：

$$K(x,x_i)=\exp\{-\frac{|x-x_i|^2}{\sigma^2}\}$$

所得的支持矢量机是一种径向基分类器，它与传统径向基函数方法的基本区别是：这里每个基函数的中心对应一个支持矢量，它们及输出权值都是由算法自动确定的。径向基形式的内积函数类似人的视觉特性，在实际应用中经常用到，需要注意的是，选择不同的 S 参数值，相应的分类面会有很大

差别。

③ S 形核函数：

$$K(\boldsymbol{x}, x_i) = \tanh[v(\boldsymbol{x} \cdot x_i) + c]$$

这时的支持矢量机算法中包含了一个隐层的多层感知器网络，网络的权值和网络的隐层节点数都是由算法自动确定的，不像传统的感知器网络那样由人凭借经验确定。此外，该算法不存在困扰神经网络的局部极小点的问题。

在上述几种常用的核函数中，最为常用的是多项式核函数和径向基核函数。除了上面提到的三种核函数外，还有指数径向基核函数、小波核函数等其他核函数，应用相对较少。事实上，需要进行训练的样本集各式各样，核函数也各有优劣。Bacsens 和 Viaene 等人曾利用 LS-SVM 分类器，采用 UCI 数据库，对线性核函数、多项式核函数和径向基核函数进行了实验比较，从实验结果来看，对不同的数据库，不同的核函数各有优劣，而径向基核函数在多数数据库上得到略为优良的性能。

3. 人工神经网络

人工神经网络[89, 90]是并行分布式系统，采用了与传统人工智能和信息处理技术完全不同的机理，克服了传统的基于逻辑符号的人工智能在处理直觉、非结构化信息方面的缺陷，具有自适应、自组织和实时学习的特点。多层感知机（Multilayer Perception，MLP）[90]是一种最常用的人工神经网络结构，本书选择其为人工神经网络的代表，并选取属性数和类别数的均值作为隐层数目。

（1）神经网络的基本原理

因为人工神经网络是人和动物的神经网络的某种结构和功能的模拟，所以要了解神经网络的工作原理，首先要了解生物神经元。其结构如图 2-5 所示。

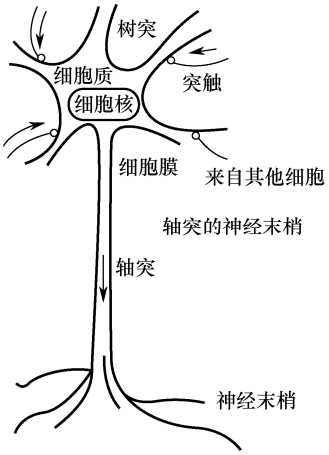


图 2-5　神经元结构

从图 2-5 可看出，生物神经元包括：①细胞体，由细胞核、细胞质与细胞膜组成；②轴突，是从细胞体向外伸出的细长部分，也就是神经纤维，轴突是神经细胞的输出端，通过它向外传出神经冲动；③树突，是细胞体向外伸出的许多较短的树枝状分支，是细胞的输入端，接收来自其他神经元的冲动；④突触，是神经元之间相互连接的地方，即神经末梢与树突相接触的交界面。

对于从同一树突先后传入的神经冲动，以及同一时间从不同树突输入的神经冲动，神经细胞均可加以综合处理，处理的结果可使细胞膜电位升高；当膜电位升高到一定阈值（约 40mV），细胞进入兴奋状态，产生神经冲动，并由轴突输出神经冲动；当输入的冲动减小，综合处理的结果使膜电位下降，当下降到阈值时，细胞进入抑制状态，此时无神经冲动输出。在任何时刻，神经细胞呈现“兴奋”状态或“抑制”状态，二者必具其一。

突触界面具有脉冲/电位信号转换功能，类似于 D/A 转换功能。沿轴突和树突传递的是等幅、恒宽、编码的离散电脉冲信号。细胞中的膜电位是连续的模拟量。

神经冲动信号的传导速度为 1～150m/s，随纤维的粗细、髓鞘的有无而

不同。

神经细胞的重要特点是具有学习功能并有遗忘和疲劳效应。总之，随着对生物神经元的深入研究，揭示出神经元不是简单的双稳逻辑元件而是微型生物信息处理机制和控制机。

神经网络的基本原理是对生物神经元进行尽可能的模拟，当然，以目前的理论水平、制造水平和应用水平，其与人脑神经网络还存在很大的差别，它只是对人脑神经网络有选择的、单一的、简化的构造和性能模拟，从而形成的不同功能、多种类型、不同层次的神经网络模型。

（2）多层感知神经网络

目前，在这一基本原理上已发展了几十种神经网络，如 Hopfield 模型、Feldmann 等的连接型网络模型、Hinton 等的玻尔兹曼机模型、Rumelhart 等人的多层感知机模型和 Kohonen 的自组织网络模型等。在众多神经网络模型中，应用最广泛的是多层感知机神经网络。

多层感知机神经网络的研究始于 20 世纪 50 年代，但一直进展不大。直到 1985 年，Rumelhart 等人提出了误差反向传递学习算法（即 BP 算法），实现了 Minsky 的多层网络设想，BP 神经网络模型如图 2-6 所示。它可以分为输入层、影层（也叫中间层）和输出层。其中，中间层可以是一层，也可以是多层，视实际情况而定。

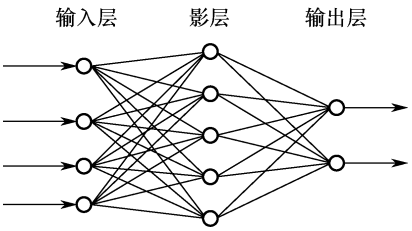


图 2-6　BP 神经网络模型

BP 网络的原理是把一个输入矢量经过影层变换成输出矢量，实现从输入空

间到输出空间的映射。由权重实现正向映射，利用当前权重作用下网络的输出与希望实现的映射要求的期望输出进行比较来学习。为减少总误差，网络利用实际误差调整权重。BP 网络必须要求与输入相对应的希望输出构成训练模式队，因而需要指导学习。BP 网络在结构上具有对称性，网络中的每个输出处理元件基本具有相同的传递函数。

BP 算法不仅有输入层节点、输出层节点，还有一个或多个隐含层节点。对于输入信号，要先向前传播到隐含层节点，经作用函数后，再把隐节点的输出信号传播到输出节点，最后给出输出结果。节点的作用的激励函数通常选取 S 形函数。该算法的学习过程由正向传播和反向传播组成。在正向传播过程中，输入信息从输入层经隐含层逐层处理，并传向输出层。每层神经元的状态只影响下一层神经元的状态。如果输出层得不到期望的输出，则转入反向传播，将误差信号沿原来的连接通道返回，通过修改各层神经元的权值，使得误差信号最小。

BP 模型把一组样本的 I/O 问题转化为一个非线性优化问题，它使用的是优化中最普通的梯度下降法。如果把神经网络看成输入到输出的映射，则这个映射是一个高度非线性映射。

设计一个神经网络的重点是模型的构成和学习算法的选择。一般来说，结构是根据所研究领域及要解决的问题确定的。通过对所研究问题的大量历史资料数据的分析及目前的神经网络理论发展水平，建立合适的模型，并针对所选的模型采用相应的学习算法，在网络学习过程中，不断地调整网络参数，直到输出结果满足要求。

4. k–最近邻分类

（1）k–最近邻分类的基本思想

k–最近邻分类算法概括来说，就是已知一个样本空间里的部分样本分成几个类，然后给定一个待分类的数据，通过计算找出与待分类数据最接近的 k 个样本，

由这 k 个样本投票决定待分类数据归为哪一类。k–最近邻分类算法在类别决策时，只与极少量的相邻样本有关。由于 k–最近邻分类算法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别，因此，对于类域的交叉或重叠较多的待分类样本集来说，k–最近邻分类算法较其他方法更为适合。

图 2-7 描述了 k–最近邻分类算法的基本思想。

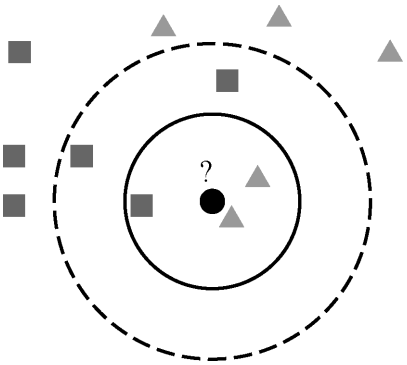


图 2-7　k–最近邻分类算法的思想

图 2-7 中有两个类型的样本数据：一类是正方形；另一类是三角形。圆形是待分类的数据。

如果 k=3，那么最近的有 2 个三角形和 1 个正方形，这 3 个点投票，则待分类点属于三角形；如果 k=5，那么圆形最近的有 2 个三角形和 3 个正方形，这 5 个点投票，则待分类点属于正方形的分类。

（2）相似性度量

样本的相似性由空间内两个点的距离来度量。距离越大，表示两个点越不相似。距离的选择有很多，常用的有如下几个。

① 欧几里得距离：

$$d_{\mathrm{EUC}}(x,y)=\left[\sum_{j=1}^{d}(x_j-y_j)^2\right]^{\frac{1}{2}}=[(x-y)(x-y)^{\mathrm{T}}]^{\frac{1}{2}}$$

马氏距离：马氏距离能够缓解由于属性的线性组合带来的距离失真，Σ是数据的协方差矩阵。

$$d_{\mathrm{MAH}}(x,y)=\sqrt{(x-y)\Sigma^{-1}(x-y)^{\mathrm{T}}}$$

② 曼哈顿距离：

$$d_{\mathrm{MAN}}(x,y)=\sum_{j=1}^{d}\left|x_j-y_j\right|$$

③ 切比雪夫距离：

$$d_{\mathrm{CHEB}}(x,y)=\max_j(|x_j-y_j|)$$

闵氏距离：r 取值为 2 时，为曼哈顿距离；r 取值为 1 时，为欧几里得距离。

$$d_{\mathrm{MIN}}(x,y)=\left(\sum_{j=1}^{d}(x_j-y_j)^r\right)^{\frac{1}{r}},r\geqslant 1$$

④ 平均距离：

$$d_{\mathrm{AVE}}(x,y)=\left[\frac{1}{d}\sum_{j=1}^{d}(x_j-y_j)^2\right]^{\frac{1}{2}}$$

⑤ 弦距离：$\|\cdot\|_2$ 表示 2-范数，即 $\|x\|_2=\sqrt{\sum_{j=1}^{d}x_j^2}$ 。

$$d_{\mathrm{CHORD}}(x,y)=\left(2-2\frac{\sum_{j=1}^{d}x_jy_j}{\|x\|_2\cdot\|y\|_2}\right)^{\frac{1}{2}}$$

⑥ 测地距离：

$$d_{\mathrm{GEO}}(x,y)=\arccos\left(1-\frac{d_{\mathrm{chord}}(x,y)}{2}\right)$$

5. 决策树

决策树[91]是最常见的分类器之一。其优点很多，包括易于理解和实现、数据准备简单、可同时处理连续和离散属性、构建过程高效等。本书选择 C4.5 算法作为决策树的代表。

决策树分类的目标是构建一个根据若干条件变量的值来预测决策变量的值的模型。该模型的形式为树结构。树的每个内部节点对应一个条件变量；该变量的每个可能取值对应一条到其子节点的边。每个叶节点表示一个对应从根节点到该叶节点路径上的条件变量值的决策变量的值。一个经典决策树如图 2-8 所示。

一个决策树可通过基于属性值检测的数据集分裂学习得到。这一过程在每个得到的子数据集上以一种称为迭代分区的迭代模式不断重复运行。迭代过程结束的条件有两个：一是在某个节点上的子数据集有相同的决策属性值；二是分裂不再增加新的预测值。这一自顶向下的决策树推断过程是贪心算法的一个例子，也是迄今为止构建决策树的最常见策略。

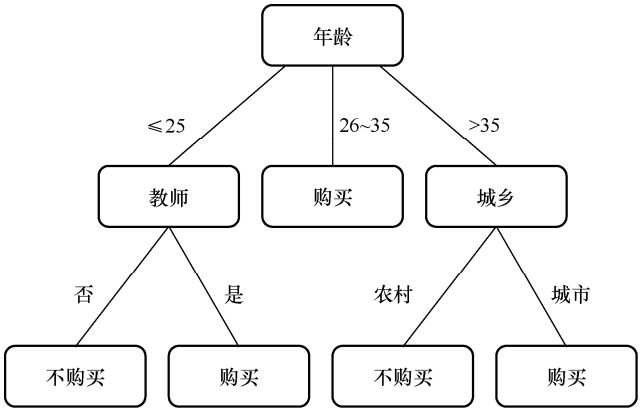


图 2-8　经典决策树

为了分类一个未知样本，其属性值被决策树不断测试。测试是沿一条自树根到包含该样本的类别预测的叶节点的路径进行的。决策树可以很容易地被转化为若干决策规则。以下内容描述了一个基本的决策树学习算法。

输入：D：训练样本集，attributeList：候选属性集（属性列表）

输出：一棵决策树

```
1.  构造一个节点 N
2.  if D 中的所有样本的类标都相同（记为 C 类）
3.      将节点 N 作为叶子节点标记为 C
4.      return 节点 N
5.  if attributeList 为空
6.      将节点 N 作为叶子节点标记为 D 中类标最多的类
7.      return 节点 N
8.  选择最佳的分裂准则 splitCriterion
9.  将节点 N 标记为最佳分裂准则 splitCriterion
10. if 分裂属性取值是离散的，并且允许决策树进行多叉分裂
11.     从属性列表中减去分裂属性
12. for each 分裂属性的每个取值 j
13.     记 D 中满足 j 的记录集合为 Dj
14.     if 如果 Dj 为空
15.         新建一个叶子节点 F，标记为 D 中类标最多的类，并且把节点 F 挂在 N 下
16.     else
17.         递归调用 GenerateDecisionTree（Dj, attributeList）得到子树节点 Nj
18.         将 Nj 挂在 N 下
19. return 节点 N
```

6. 随机森林

为了克服决策树容易过度拟合的缺点，Breiman 提出了一种新的组合分类器算法——随机森林算法[86]。他把分类决策树组合成随即森林，即在变量（列）的使用和数据（行）的使用上进行随机化，生成很多分类树，再汇总分类树的

结果。随机森林在运算量没有显著提高的前提下提高了预测精度，对多元共线性不敏感，是当前最好的算法之一。

（1）随机森林的基本思想

随机森林是通过自助法（Bootstrap）重复采样技术，从原始训练样本集 S 中有放回地重复随机抽取 k 个样本，生成新的训练集样本集合，然后根据自助样本生成 k 个决策树组成的随机森林。其实质是对决策树算法的一种改进，将多个决策树合并在一起，每棵树的建立依赖一个独立抽取的样本，森林中的每棵树具有相同的分布，分类误差取决于每棵树的分类能力和它们之间的相关性。

根据随机森林的原理和基本思想，随机森林的生成主要包括以下三个步骤：

首先，通过 Bootstrap 方法在原始样本集 S 中抽取 k 个训练样本集，一般情况下，每个训练集的样本容量与 S 一致。

其次，对 k 个训练集进行学习并生成 k 个决策树模型。在决策树生成过程中，假设共有 M 个输入变量，从 M 个变量中随机抽取 F 个变量，各个内部节点均利用这 F 个特征变量上最优的分裂方式来分裂，且 F 值在随机森林模型的形成过程中为恒定常数。

最后，将 k 个决策树的结果进行组合，形成最终结果。针对分类问题，组合方法是简单多数投票法；针对回归问题，组合方法则是简单平均法。

（2）随机森林的重要参数

① 随机森林中单棵树的分类强度和任意两棵树间的相关度。在随机森林中，每棵决策树的分类强度越大，即每棵树枝叶越茂盛，则整体随机森林的分类性能越好；树与树之间的相关度越大，即树与树之间的枝叶相互穿插越多，则随机森林的分类性能越差。减少树之间的相关度可以有效地降低随机森林的总体误差率，同时增加每棵决策树的强度。因为它是由 Bootstrap 方法来形成训练集的，随机抓取特征来分裂，并且不对单棵树进行剪枝，使得随机森林模型

能够具有较高的噪声容忍度和较大的分类强度，同时也降低了任意两棵树之间的相关度。

② OOB（Out Of Bag）估计。应用 Bootstrap 方法时，在原始样本集 S 中进行 k 次有放回的简单随机抽样，形成训练样本集。在使用 Bootstrap 对 S 进行抽样时，每个样本未被抽取的概率 p 为 $(1-1/n)^n$。当 n 足够大时，p=0.368，表明原始样本集 S 中接近 37%的样本不会出现在训练样本集中，这些未被抽中的样本称为 OOB。利用这部分样本进行模型性能的估计称为 OOB 估计，这种估计方法类似于交叉验证。在随机分类模型中，它是分类模型的出错率；在随机回归模型中，它是回归模型的残差。

③ 对模型中变量重要性的估计。随机森林计算变量重要性有两种方法：一种是基于 OOB 误差的方法，称为 MDA（Mean Decrease Accuracy）；另一种是基于 Gini 不纯度的方法，称为 MDG（Mean Decrease Gini）。两种方法都是下降越多表示变量越重要。

MDA 具体描述如下：

第一，训练随机森林模型，利用袋外样本数据测试模型中每棵树的 OOB 误差。

第二，随机打乱袋外样本数据中变量 v 的值，重新测试每棵树的 OOB 误差。

第三，两次测试的 OOB 误差的差值的平均值，即为单棵树对变量 v 重要性的度量值，计算公式为

$$\mathrm{MDA}(v)=\frac{1}{\mathrm{ntree}}\sum_{t}\left(\mathrm{errOOB}_t-\mathrm{errOOB}'_t\right)$$

MDG 具体描述如下：基于 Gini 的变量重要性是用变量 v 导致的 Gini 不纯度的降低来衡量的。在分类节点 t，Gini 系数不纯度的计算公式为

$$G(t)=1-\sum_{K=1}^{Q}p^2\left(k/t\right)$$

其中，Q 代表目标变量的类别总数，$p(k/t)$ 代表在节点 t 中目标变量为第 k 类

的条件概率。根据公式计算每棵树的 Gini 不纯度下降值，再将所有树的结果进行平均。

7. 关联分类

关联分类[92]是基于关联规则挖掘的分类技术。首先，从所有关联规则中抽取后件是分类属性的子集。然后，对该子集进行剪枝和排序。最后，基于该约简的规则集，AC 可构建一个有效的分类器。AC 通常比 DT 有更高的准确率。本书选择常见的 L3 算法[60]作为 AC 的代表。

2.2.2 经验研究

1. 数据集

本书选择 MAROB[19]数据集作为实验数据集。数据集的属性分为两类：环境属性和行为属性。前者刻画组织所处的环境，后者表示组织当年的行为。每个数据集的样本数一般小于 100。

2. 评价指标

本书使用准确率（Accuracy）、召回率（Recall）与 AUC 值作为组织行为预测模型质量的评价指标。

混淆矩阵如表 2-2 所示，则准确率与召回率分别表示为

$$\text{Accuracy}=(TP+TN)/(TP+FP+FN+TN)$$

$$\text{Recall} = TP/(TP + FN)$$

表 2-2 混淆矩阵

	预测正样本数	预测负样本数
正样本数	正确的正样本数（TP）	错误的负样本数（FN）
负样本数	错误的正样本数（FP）	正确的负样本数（TN）

AUC 是 ROC 曲线下的面积值。无论是对平衡的还是非平衡的数据集，AUC 都是比准确率更一致且辨别力更强的评价指标[93]。

3. 实验结果与分析

表 2-3 展示了准确率、召回率与 AUC 值的均值与标准差。每个指标的最高均值与最低标准差被加粗显示。

表 2-3　准确率、召回率与 AUC 值的均值与标准差

指标	统计量	算法						
		NB	**SMO**	**MLP**	**kNN**	**C4.5**	**RF**	**AC**
Accuracy	AVG	**0.969**	0.941	0.952	0.944	0.929	0.940	0.929
	SD	**0.044**	0.060	0.053	0.052	0.069	0.056	0.063
AUC	AVG	**0.978**	0.697	0.915	0.854	0.530	0.788	0.866
	SD	**0.054**	0.216	0.168	0.249	0.401	0.227	0.154
Recall	AVG	**0.778**	0.420	0.525	0.344	0.362	0.428	0.430
	SD	**0.303**	0.434	0.409	0.405	0.404	0.404	0.431

表 2-3 表明所有算法的准确率都超过 90%，而召回率都很低。这是因为实验数据分布不均衡，也就是说，大部分数据集中负样本比例较小而正样本比例较大。类不平衡程度越高，训练集越小，则分类器对类不平衡问题越敏感[94]。另外，AUC 值依算法不同而差别很大。

朴素贝叶斯（NB）在所有指标下性能都最好，相应的方差也最小。NB 性能最佳有两个原因。一是 MAROB 数据集一般都比较小（每个组织只有几十个实例），而 NB 在比较小的数据集上一般表现非常好[95]。二是在 0-1 损失函数下，甚至当数据集中的属性并不满足非依赖性假设时，NB 也可能是最优分类器[95]。

C4.5 在所有指标下性能都很差。这是因为其在数据集比较小的情况下，容易误分类[96]。RF 在召回率和 AUC 值这两个指标下比 C4.5 好很多，这是因为集成分类器一般优于单个弱分类器。同样是基于规则的分类器，AC 的性能却明显好于 C4.5，而这和很多关于这两种方法的比较的发现（如文献[92,97,98,99]）是一致的。

k–最近邻分类（kNN）在召回率指标下的性能最差。这是因为 kNN 一般在稠密数据集上的性能较好，而对比较稀疏的数据集，其很难为待分类样本找到可信邻居[100]。另外，MAROB 数据集中正样本尤其稀少，这使得该问题对正样本更加突出。最后，相比支持矢量机（SVM）和多层感知器（MLP）等分类器，kNN 对类不平衡问题更加敏感[94, 101, 102, 103]。

2.3　代价敏感组织行为预测建模

2.2 节的实验结论指出，在本领域普遍存在的非一致误分类代价与类不平衡问题使得标准分类算法所构建的组织行为预测模型的质量严重降低。代价敏感学习是为解决非一致误分类代价问题，以取得最小或较小样本期望误分类代价为目标的机器学习方法。本节研究了四种典型代价敏感学习方法基于不同标准分类器建立的预测模型在不同情形下的性能。另外，基于上采样方法提出了一个对不同正样本采用不同复制策略的适用于本领域的高性能代价敏感算法。最后，提出了一种基于代价曲线的针对本领域类不平衡与非一致误分类代价问题的有效个性化解决方案。

2.3.1　代价敏感学习方法

本节介绍四种有代表性的代价敏感学习方法。因为在组织行为建模领域有两类情况占支配性地位，所以本节中误分类代价以 2×2 代价矩阵 $\boldsymbol{C}$ 表示，其中，$C(0,0)$和 $C(1,1)$表示正确分类的代价，其值为 0，$C(0,1)$表示负样本误分类代价，$C(1,0)$表示正样本误分类代价。为表示方便，$C(1,0)$和 $C(0,1)$分别表示为 c_0 和 c_1，N_0 和 N_1 分别表示训练集中负类和正类样本的数量。注意，在某些领域中 c_0 和 c_1 可能为主观值。

1. 上采样

上采样方法通过生成一定数量的少数类样本同时保持多数类样本数目不变来平衡训练集的类分布。少数类样本的生成遵从特定的规则。最常见的规则是

随机复制少数类样本。因为上采样引入了额外的训练样本，所以分类器的构建会花费较多的时间。然而，这一缺点在本领域可以忽略，因为组织行为数据集（如 MAROB）一般规模较小。另外，复制正类样本会导致过拟合[31]。

上采样一般用来处理类不平衡问题。它可以通过关联采样率与 c_1 和 c_0 的比率成为代价敏感学习方法[29]。也就是说，为做出最优决策，上采样之后正类样本的数目应根据下式计算：

$$N_1' = (c_1/c_0)N_1$$

其中，N_1' 是上采样后正类数目。

2. 下采样

作为处理类不平衡问题的常用方法，下采样通过删减一定数量的多数类样本同时保持少数类样本数目不变来平衡训练集的类分布。下采样可以采用随机策略或确定性策略。文献[36]建议将下采样方法作为基准方法用于算法比较。然而，下采样可能因为损失信息导致性能不佳，当数据集很小时尤其如此。下采样方法可以通过关联采样率与 c_0 和 c_1 的比率成为代价敏感学习方法[29]。也就是说，为做出最优决策，下采样之后负类样本的数目应根据下式计算：

$$N_0' = (c_0/c_1)N_0$$

其中，N_0' 是下采样后负类数目。

相比随机下采样，确定下采样可以减少方差，但却因删减的样本可能包含对学习过程潜在有用的信息而增加了引入偏差的风险[29]。堆叠泛化方法[104]可同时减少方差和偏差。其主要过程可描述如下。首先，从多数类样本中随机采样几个子集；其次，用每个子集训练一个弱分类器；最后，对所有弱分类器输出的类概率取平均值。

3. MetaCost

MetaCost 方法[27]可通过对任一分类器包装一个最小化代价的过程而使其变

得代价敏感。其过程可简要描述如下。首先，使用 bagging 方法构建多分类器的一个集成；其次，根据代价比和集成输出估计的最优类重标号训练样本；最后，基于重标号的数据集构建分类器。一方面，MetaCost 方法在很多基准数据集上表现出很好的有效性和可伸缩性[27]；另一方面，不管用 bagging 还是 boosting 作为内部分类器，在多数情况下，MetaCost 方法表现不佳（如文献[105]）。

4. 调整决策阈值

调整决策阈值法（Threshold-moving）把决策阈值向分类代价较高的类移动以使正确识别该类样本变得容易。分类阈值改变后，样本的分类将使期望误分类代价最小。为做出最优预测，新的决策阈值应根据下式计算[29]：

$$T = c_0 / (c_0 + c_1)$$

调整类概率估计也可以达到调整决策阈值一样的效果。假定 $O_i (i \in \{0,1\})$ 表示少数类和多数类的初始概率估计，调整后的类概率估计根据下式计算[30]：

$$O_i' = O_i c_i \Big/ \sum_{j=0}^{1} O_j c_i, i \in \{0,1\}$$

调整决策阈值法是最直接的代价敏感学习方法，因为它并不改变分类器的学习过程。相比其他的代价敏感学习方法，当误分类代价改变时，该方法不需要对分类模型做任何改变。然而，该方法仅当基分类器生成准确的类概率时才有较好表现。像 NB，ANN，DT 和 RF 这样的分类器不用做任何改变就可以生成类概率估计。kNN 可以将样本属于类 c 的概率估计为该样本 k 个最近邻中属于类 c 的样本的比例。因为 SVM 不能产生类概率估计，所以不能应用该方法。

2.3.2　经验研究

1. 实验设置

实验数据集采用 MAROB 数据集。实验选取了类不平衡程度为 0.478～

0.077 的 15 个子数据集。基分类器选取应用最广泛的 NB、SVM、ANN、kNN、DT 和 RF。考虑领域特点和专家建议，c_1 和 c_0 的比率在区间[1, 10]中选取。期望误分类代价作为评价指标。

2. 数据集

假如要预测组织 *G* 的行为 *B*，则抽取所有环境属性和行为属性 *B* 作为实验用数据集并命名为 *G-B*。每个数据集的样本数一般小于 100（这里，PFLP、FRC、DFLP、FU 和 PPS 分别表示 Popular Front for the Liberation of Palestine、Fatah Revolutionary Council、Democratic Front for the Liberation of Palestine、Fatah the Uprising 和 Palestinian Popular Struggle Front）。

数据集的详细信息如表 2-4 所示。

表 2-4　数据集信息

编号	数据集	样本量	正样本比率/%
1	PFLP-ARMATTACK	46	47.8
2	FRC-ARMATTACK	25	44.0
3	DFLP-ARMATTACK	60	30.0
4	FU-ARMATTACK	27	29.6
5	FRC-BOMB	25	24.0
6	HAMAS-ARSON	46	17.4
7	HAMAS-ASSASSIN	46	15.2
8	HAMAS-KIDNAP	46	15.2
9	FU-BOMB	27	14.8
10	PPSF-ARMATTACK	36	13.9
11	PFLP-ASSASSIN	46	13.0
12	DFLP-BOMB	60	10.0
13	PFLP-SUICIDE	46	8.7
14	Fatah-KIDNAP	65	7.7
15	Fatah-SUICIDE	65	7.7

3. 评价指标

本节选择期望误分类代价作为评价指标，其定义为

$$E[C] = p(+)c_1\mathrm{FN} + (1 - p(+))\mathrm{FP}$$

其中，FP，FN 和 $p(+)$分别表示错误的正类比率、错误的负类比率和正类比率。本书使用留一交叉验证法获取 FP 和 FN。留一交叉验证法理论上可获得对混淆矩阵的最优估计。其主要缺点是大量训练过程导致的高计算代价。因为在本领域中数据集通常很小，所以该缺点可被忽略。c_0 取 1，c_1 分别取区间[1, 10]内的非重复整数，就可在相同的训练集和测试集上得到 10 个结果。最终试验结果取 $10\times K$ 个结果的均值。

4. 实验结果

表 2-5 展示了分类器、代价敏感学习方法的实验结果与比率。图 2-9 展示了方法−分类器组合、分类器与代价敏感学习方法在 15 个数据集上的平均结果。图 2-10 展示了方法−分类器组合、分类器与代价敏感学习方法在正类比率小于 15%的极度不平衡的数据集上的平均结果。图 2-11 展示了代价敏感学习方法与相应基分类器的实验结果比率。x 轴表示正类比率。不同的直线拟合了对应相同方法与分类器的结果点。

表 2-5　标准分类器、代价敏感学习方法的实验结果与比率

数据集编号	单分类器	上采样		下采样		MetaCost		调整决策阈值法	
	代价	代价	比率	代价	比率	代价	比率	代价	比率
NB									
1	1.67	0.56	0.335	0.49	0.293	0.46	0.275	0.46	0.275
2	1.76	0.43	0.244	0.51	0.290	0.53	0.301	0.53	0.301
3	1.24	0.48	0.387	0.47	0.379	0.46	0.371	0.48	0.387
4	1.22	0.65	0.533	0.67	0.549	0.76	0.623	0.75	0.615
5	1.32	0.52	0.394	0.68	0.515	0.74	0.561	0.74	0.561
6	0.87	0.15	0.172	0.15	0.172	0.15	0.172	0.15	0.172
7	0.18	0.25	1.389	0.23	1.278	0.29	1.611	0.28	1.556
8	0.30	0.36	1.200	0.30	1.000	0.33	1.100	0.38	1.267
9	0.89	0.33	0.371	0.34	0.382	0.30	0.337	0.33	0.371
10	0.79	0.44	0.557	0.47	0.595	0.46	0.582	0.44	0.557
11	0.23	0.25	1.087	0.27	1.174	0.25	1.087	0.25	1.087

（续表）

数据集编号	单分类器	上采样		下采样		MetaCost		调整决策阈值法	
	代价	代价	比率	代价	比率	代价	比率	代价	比率
NB									
12	0.55	0.33	0.600	0.40	0.727	0.38	0.691	0.36	0.655
13	0.14	0.23	1.643	0.22	1.571	0.26	1.857	0.27	1.929
14	0.13	0.14	1.077	0.16	1.231	0.18	1.385	0.16	1.231
15	0.48	0.31	0.646	0.32	0.667	0.29	0.604	0.31	0.646
MLP									
1	1.67	0.47	0.281	0.47	0.281	0.47	0.281	0.46	0.275
2	1.76	0.43	0.244	0.51	0.290	0.46	0.261	0.44	0.250
3	1.21	0.45	0.372	0.46	0.380	0.46	0.380	0.45	0.372
4	1.22	0.61	0.500	0.65	0.533	0.64	0.525	0.56	0.459
5	1.32	0.49	0.371	0.66	0.500	0.56	0.424	0.51	0.386
6	0.80	0.17	0.213	0.17	0.213	0.16	0.200	0.17	0.213
7	0.18	0.25	1.389	0.22	1.222	0.26	1.444	0.24	1.333
8	0.42	0.23	0.548	0.26	0.619	0.23	0.548	0.22	0.524
9	0.44	0.51	1.159	0.35	0.795	0.38	0.864	0.51	1.159
10	0.79	0.59	0.747	0.53	0.671	0.57	0.722	0.60	0.759
11	0.74	0.37	0.500	0.38	0.514	0.35	0.473	0.32	0.432
12	0.55	0.34	0.618	0.39	0.709	0.37	0.673	0.36	0.655
13	0.14	0.14	1.000	0.22	1.571	0.14	1.000	0.14	1.000
14	0.47	0.19	0.404	0.18	0.383	0.19	0.404	0.17	0.362
15	0.44	0.35	0.795	0.40	0.909	0.28	0.636	0.29	0.659
SVM									
1	1.43	0.46	0.322	0.46	0.322	0.52	0.364	1.43	1.000
2	1.43	0.46	0.322	0.46	0.322	0.52	0.364	1.43	1.000
3	1.21	0.44	0.364	0.45	0.372	1.21	1.000	1.21	1.000
4	1.22	0.62	0.508	0.63	0.516	1.28	1.049	1.22	1.000
5	1.32	0.44	0.333	0.66	0.500	1.32	1.000	1.32	1.000
6	0.72	0.15	0.208	0.19	0.264	0.41	0.569	0.72	1.000
7	0.26	0.20	0.769	0.23	0.885	0.26	1.000	0.26	1.000
8	0.42	0.20	0.476	0.31	0.738	0.42	1.000	0.42	1.000
9	0.81	0.29	0.358	0.47	0.580	0.67	0.827	0.81	1.000

（续表）

数据集	单分类器	上采样		下采样		MetaCost		调整决策阈值法	
编号	代价	代价	比率	代价	比率	代价	比率	代价	比率
SVM									
10	0.76	0.54	0.711	0.59	0.776	0.76	1.000	0.76	1.000
11	0.72	0.27	0.375	0.31	0.431	0.26	0.361	0.72	1.000
12	0.55	0.38	0.691	0.41	0.745	0.55	1.000	0.55	1.000
13	0.14	0.14	1.000	0.14	1.000	0.14	1.000	0.14	1.000
14	0.42	0.19	0.452	0.20	0.476	0.42	1.000	0.42	1.000
15	0.42	0.28	0.667	0.40	0.952	0.42	1.000	0.42	1.000
C4.5									
1	1.67	0.46	0.275	0.46	0.275	0.46	0.275	0.46	0.275
2	1.76	0.54	0.307	0.54	0.307	0.55	0.313	0.54	0.307
3	1.21	0.47	0.388	0.51	0.421	0.64	0.529	0.61	0.504
4	1.63	0.74	0.454	0.62	0.380	0.70	0.429	0.69	0.423
5	1.32	0.66	0.500	0.73	0.553	0.75	0.568	0.68	0.515
6	1.11	0.17	0.153	0.22	0.198	0.17	0.153	0.23	0.207
7	0.28	0.26	0.929	0.27	0.964	0.25	0.893	0.26	0.929
8	0.90	0.20	0.222	0.23	0.256	0.24	0.267	0.37	0.411
9	0.89	0.30	0.337	0.33	0.371	0.30	0.337	0.59	0.663
10	0.76	0.48	0.632	0.65	0.855	0.76	1.000	0.73	0.961
11	0.72	0.40	0.556	0.34	0.472	0.34	0.472	0.34	0.472
12	0.55	0.40	0.727	0.44	0.800	0.62	1.127	0.73	1.327
13	0.50	0.15	0.300	0.27	0.540	0.18	0.360	0.50	1.000
14	0.47	0.20	0.426	0.19	0.404	0.20	0.426	0.47	1.000
15	0.42	0.29	0.690	0.41	0.976	0.48	1.143	0.42	1.000
kNN									
1	1.67	0.48	0.287	0.48	0.287	0.48	0.287	0.48	0.287
2	1.76	0.55	0.313	0.55	0.313	0.56	0.318	0.55	0.313
3	1.21	0.48	0.397	0.64	0.529	0.45	0.372	0.62	0.512
4	1.22	0.63	0.516	0.66	0.541	0.64	0.525	0.62	0.508
5	1.32	0.68	0.515	0.77	0.583	0.69	0.523	0.73	0.553

（续表）

数据集	单分类器	上采样		下采样		MetaCost		调整决策阈值法	
编号	代价	代价	比率	代价	比率	代价	比率	代价	比率
kNN									
6	0.72	0.18	0.250	0.55	0.764	0.16	0.222	0.18	0.250
7	0.16	0.17	1.063	0.62	3.875	0.26	1.625	0.20	1.250
8	0.28	0.24	0.857	0.66	2.357	0.61	2.179	0.37	1.321
9	0.81	0.39	0.481	0.84	1.037	0.81	1.000	0.56	0.691
10	0.76	0.50	0.658	0.73	0.961	0.54	0.711	0.58	0.763
11	0.72	0.25	0.347	0.82	1.139	0.26	0.361	0.27	0.375
12	0.55	0.39	0.709	0.62	1.127	0.40	0.727	0.35	0.636
13	0.14	0.23	1.643	0.48	3.429	0.48	3.429	0.51	3.643
14	0.34	0.21	0.618	0.52	1.529	0.38	1.118	0.16	0.471
15	0.42	0.28	0.667	0.42	1.000	0.41	0.976	0.42	1.000
RF									
1	1.67	0.46	0.275	0.47	0.281	0.50	0.299	0.47	0.281
2	1.80	0.46	0.256	0.53	0.294	0.50	0.278	0.49	0.272
3	1.21	0.46	0.380	0.46	0.380	0.43	0.355	0.43	0.355
4	1.22	0.72	0.590	0.71	0.582	0.80	0.656	0.75	0.615
5	1.32	0.52	0.394	0.67	0.508	0.60	0.455	0.58	0.439
6	0.80	0.16	0.200	0.20	0.250	0.17	0.213	0.17	0.213
7	0.16	0.23	1.438	0.25	1.563	0.24	1.500	0.25	1.563
8	0.42	0.20	0.476	0.23	0.548	0.21	0.500	0.19	0.452
9	0.48	0.40	0.833	0.40	0.833	0.30	0.625	0.32	0.667
10	0.79	0.62	0.785	0.57	0.722	0.60	0.759	0.61	0.772
11	0.74	0.40	0.541	0.33	0.446	0.40	0.541	0.40	0.541
12	0.55	0.39	0.709	0.43	0.782	0.38	0.691	0.38	0.691
13	0.14	0.14	1.000	0.22	1.571	0.14	1.000	0.14	1.000
14	0.30	0.18	0.600	0.19	0.633	0.19	0.633	0.19	0.633
15	0.44	0.27	0.614	0.35	0.795	0.27	0.614	0.28	0.636

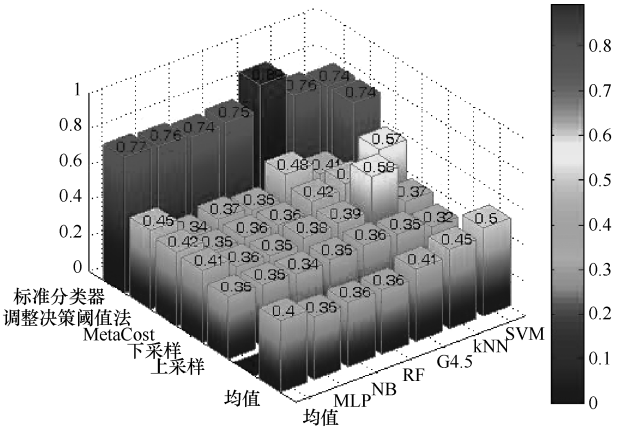


图 2-9　所有数据集上的平均实验结果

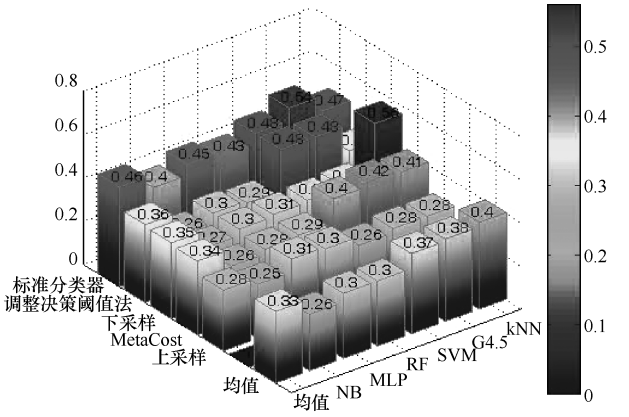
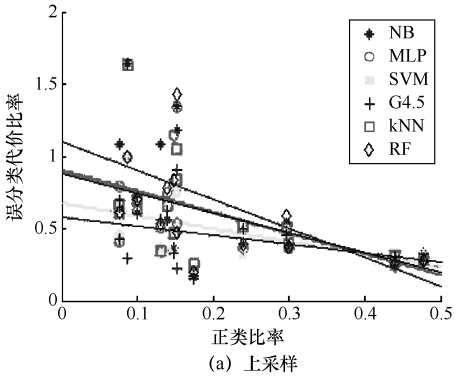
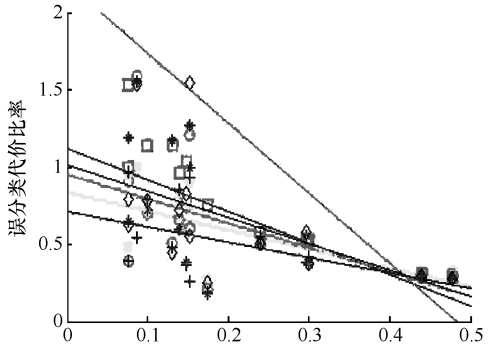


图 2-10　高度不平衡数据集上的平均实验结果

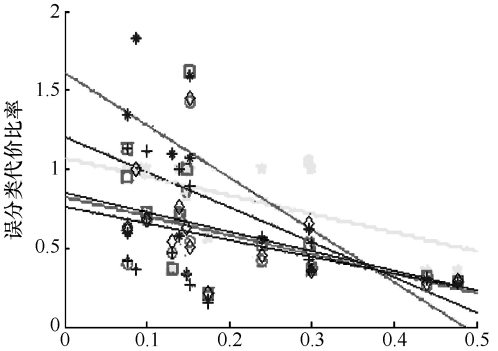


(a) 上采样

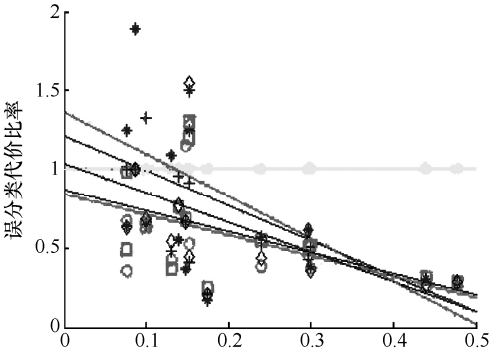
图 2-11　代价敏感学习方法与相应分类器的实验结果比率



(b) 下采样



(c) MetaCost



(d) 调整决策阈值法

图 2-11　代价敏感学习方法与相应分类器的实验结果比率（续）

5. 观察与分析

（1）对在所有数据集上的实验结果（图 2-9 和表 2-5）的观察与分析

① 大部分方法−分类器组合比相应的基分类器减少了大约 50%的期望误分类代价。这表明不管采用何种基分类器（除了 SVM），这四种代价敏感学习方法都表现出很好的性能。文献[26]建立了样本分布、每类先验概率、每类误分类代价与决策阈值之间的联系。这一联系表明这四种方法在理论上是等价的。另外，这四种方法的性能有着或多或少的差异，这是因为它们之间的精确联系是复杂的且依赖于具体任务/方法[33]的。文献[32, 94]在其他领域也得到了类似的发现。

② 上采样取得了比其他三种方法好得多的平均结果。其在除 MLP 外的所有分类器上都明显取得了最好的结果。下采样方法取得了次优的平均结果。这一发现与大多数关于上采样和下采样性能比较的发现（如文献[27, 31, 106]）矛盾。原因是下采样会使数据集中的信息匮乏问题变得更加突出[106]，尽管堆叠泛化方法被用来改善下采样的信息损失问题。另外，MetaCost 将一些负类样本重标号为正类，从而使一些潜在有用的信息丢失。这或许是 MetaCost 和下采样表现相当的原因。最后，调整决策阈值法表现最差，主要原因是其不能应用于 SVM。

③ MLP 取得了最好的平均结果。NB 和 RF 的性能与 MLP 相差不多。kNN 表现不佳是因为稀疏的数据集很难形成可信的邻近样本[96]。下采样使得该问题更加突出，而这使得下采样−kNN 组合在所有方法与 kNN 的组合中表现最差。Threshold-moving-C4.5 表现较差是因为 C4.5 较差的类概率估计能力[107]，而这使得 C4.5 的性能在平均值以下。基于 RF 的方法比基于 C4.5 的方法平均性能好，因为 RF 本身就比 C4.5 分类准确率高。

（2）对在高度不平衡数据集上的实验结果（图 2-10 和表 2-5）的观察与分析

① 方法−分类器组合比相应的基分类器平均减少了约 30%的期望误分类代价。这表明总体上这四种代价敏感学习方法是很有效的。这一发现与其他很多处理类不平衡和非一致误分类代价问题的研究的发现是一致的（如文献[27,

31, 108])。

② 上采样方法取得了比其他三种方法突出得多的平均结果。这是因为在高度不平衡数据集中正样本特别稀少，而上采样是唯一直接解决绝对稀少问题的方法[108]。

③ 对每种方法，NB 都取得了最佳的平均结果，而 MLP 和 RF 以 0.4 的距离紧随其后。这是因为本领域数据集通常很小，而且正样本的严重稀少使该问题对其他分类器的影响更加严重[95]。

④ 下采样−kNN 组合的效果甚至比 kNN 都差，而 Threshold-moving-C4.5 组合相比 C4.5 优势很小。除了上面提及的原因，还因为 kNN 和 C4.5 对类不平衡问题比较敏感[94, 103]，而下采样−kNN 组合和 Threshold-moving-C4.5 组合使该问题更加突出[83]。

（3）对四种代价敏感学习方法与相应基分类器的结果比值（图 2-11 和表 2-5）的观察和分析

① 一般来讲，对所有方法−分类器组合（除 Threshold-moving-SVM），正类比率和代价敏感学习方法与相应基分类器的结果比值成反比。这表明高的类不平衡程度阻碍代价敏感学习方法的性能，而这与文献[30]中的发现一致。误分类代价较大的样本越少，代价敏感学习方法对约减期望误分类代价的贡献越小。

② 类不平衡问题对基于 C4.5 的代价敏感学习方法的影响较小，而对基于 kNN 的代价敏感学习方法的影响较大。这是因为本领域数据分布比较稀疏而 kNN 的学习过程依赖较平衡的类分布以使邻近样本更可信。

下面简要总结以上内容。首先，上采样方法是解决组织行为预测建模中类不平衡问题与非一致误分类代价问题的较优代价敏感学习方法。另外，总体上，MLP、NB 和 RF 都是基分类器的较好选择，而在类高度不平衡条件下，NB 是最好的选择。尽管大部分方法−分类器组合总体上都是有效的，但高的类不平衡程度将阻碍代价敏感学习方法的性能。

2.3.3　OESP 算法

2.3.2 节中的实验结果表明上采样是本领域中比较好的代价敏感学习方法。然而，该方法的一个主要缺点是过拟合，本领域数据集的稀疏性与不平衡类分布使得问题更加严重。

定义 2-1　假定 p 为正样本，N 为 p 的近邻集合，P 是 N 的正样本子集，p 是孤立正样本，当且仅当 $|N|>2$，$|P|<2$。

很明显，对孤立正样本的复制会导致过拟合。如果对孤立正样本采取不同的上采样策略以减少过拟合的情况，则上采样方法的性能有望得到提高。基于这一考虑，提出了算法 OESP（Oversampling Excluding Isolated Positives），其通过正常复制非孤立正样本而保持孤立正样本不变来改进上采样方法。

以下内容描述了算法 OESP。

输入：S：原始数据集，c_0, c_1

输出：C：分类器

1. $S' \leftarrow S$
2. **for each** 正类样本 $p \in S$
3. **if** p 不是一个孤立正样本
4. $S' \leftarrow$ p 的 $(c_1/c_0 - 1)$ 份复制
5. 将 S'作为训练集训练一个 NB 分类器 C
6. **return** C

为避免过拟合，文献[33]提出了上采样方法 SMOTE，其生成合成样本而不是对样本进行复制。首先，基于需要做上采样的样本数量，对每个正样本，随机选取其 k 个最近同类邻居。然后，新的合成样本沿着该样本与选取的邻居之

间的连线生成。

SMOTE 被证明是解决类不平衡与非一致误分类代价问题的有效方法。通过关联采样率与 c_1 和 c_0 的比率可使 SMOTE 成为代价敏感学习方法。这一关系定义为

$$k=\left(c_1/c_0-1\right)$$

表 2-6 展示了 NB、上采样、OESP 和 SMOTE 的期望误分类代价和标准差。对每个数据集，性能最好的方法的期望代价被加粗显示。注意，实验设置同上节。另外，因为 NB 在不同情况下均表现优异，所以仅选择其作为基分类器。

从表 2-6 可以看出，算法 OESP 在大部分数据集上取得了比上采样和 SMOTE 更小的期望误分类代价。很明显，OESP 要优于上采样和其他方法，尤其是当类不平衡程度比较高时。另外，SMOTE 与 NB 相比优势并不明显。

表 2-6　NB、上采样、OESP 和 SMOTE 的期望误分类代价和标准差

数据集	NB	上采样	OESP	SMOTE
1	1.674±0.922	**0.559**±0.107	**0.559**±0.107	1.674±0.922
2	1.760±0.968	**0.428**±0.036	**0.428**±0.036	1.540±0.848
3	**1.242**±0.683	0.475±0.155	0.475±0.155	**1.242**±0.657
4	1.222±0.674	0.652±0.156	**0.585**±0.230	1.222±0.674
5	1.320±0.728	**0.524**±0.100	**0.524**±0.100	1.356±0.732
6	0.870±0.478	**0.152**±0.000	**0.152**±0.000	**0.152**±0.000
7	**0.185**±0.102	0.250±0.065	**0.185**±0.130	**0.185**±0.065
8	**0.304**±0.167	0.359±0.183	0.359±0.183	**0.304**±0.133
9	0.889±0.489	**0.326**±0.096	0.370±0.100	0.685±0.337
10	0.792±0.436	**0.444**±0.122	**0.444**±0.122	0.639±0.336
11	**0.228**±0.126	0.248±0.074	0.235±0.070	**0.228**±0.065
12	0.550±0.303	**0.333**±0.175	0.367±0.140	0.458±0.252
13	**0.141**±0.078	0.233±0.065	**0.141**±0.159	**0.141**±0.065
14	**0.131**±0.072	0.142±0.046	**0.131**±0.710	**0.131**±0.046
15	0.485±0.266	0.312±0.117	**0.295**±0.140	0.400±0.186
均值	0.786±0.433	0.362±0.100	**0.350**±0.116	0.691±0.355

注：加黑体的数据是该行数据中的最小值。

2.3.4　基于代价曲线的解决方案

本章分析评价了典型的代价敏感学习方法在组织行为预测建模领域的平均性能表现，以作为实际应用中代价敏感学习方法的选择依据。然而，在数据集确定的情况下，平均性能最好的代价敏感学习方法在该数据集上的表现未必最好。另外，本领域中误分类代价比不易确定且容易随时间等因素的变化而变化。为在数据集确定而误分类代价比不易确定且易变的情况下快速、灵活地为用户推荐性能最佳的代价敏感学习方法，本节提出了一个基于代价曲线[109]的解决方案。

1. 代价曲线

为清晰地表示分类模型的期望误分类代价，文献[109]提出了代价曲线（Cost Curve）的概念。代价曲线是 ROC 曲线的对偶表示，可使对分类器的期望误分类代价的比较变得很容易。

概率−代价函数 PCF 定义为

$$\mathrm{PCF}(+)=\frac{p(+)c_1}{p(+)c_1+p(-)c_0}$$

假定 $p(+)$、$p(-)$、TP、FP、PCF 分别表示给定样本 e 属于正类和负类的先验概率、正确正样本率、错误正样本率、概率−代价函数，则归一化的期望误分类代价表示为[110]

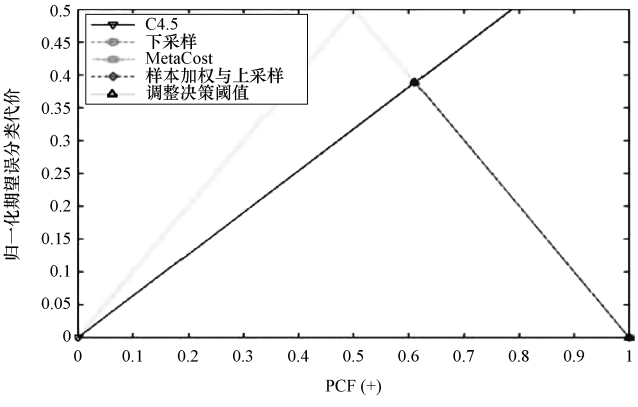
$$\mathrm{NE}[C] = (1-\mathrm{TP}-\mathrm{FP})\ \mathrm{PCF}(+)+\mathrm{FP}$$

以 PCF(+)、NE[C]分别为 x 轴和 y 轴，则用 TP 和 FP 定义的分类器的性能被表示为一条代价曲线。代价曲线下的面积是总期望代价。两条曲线下的面积大小反映了对应的两个分类器的期望性能的好坏。如果在 x 轴某范围内一个分类器的代价曲线低于另一个，则其在该范围内优于另一个分类器。

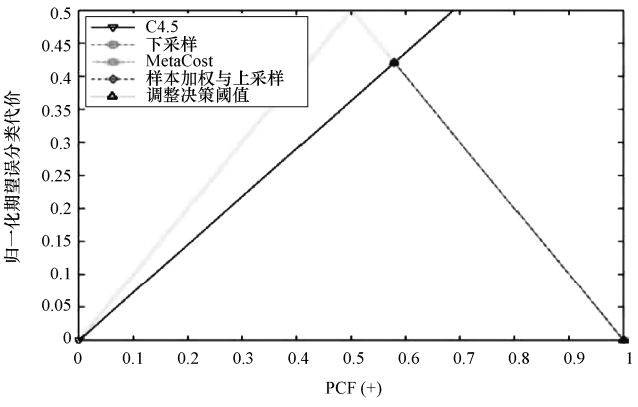
2. 实验结果

注意，实验设置及所选代价敏感学习方法同 2.2 节。另外，本节仅选择 C4.5 为基分类器。

除 Fatah-SUICIDE 外的 14 个数据集的代价曲线分别如图 2-12（a）～（n）所示。每幅图均包含 5 条代价曲线，分别对应 C4.5、下采样、MetaCost、上采样与调整决策阈值法。曲线与 x 轴所包围的区域为有效区域，可以用该区域来识别有效方法。

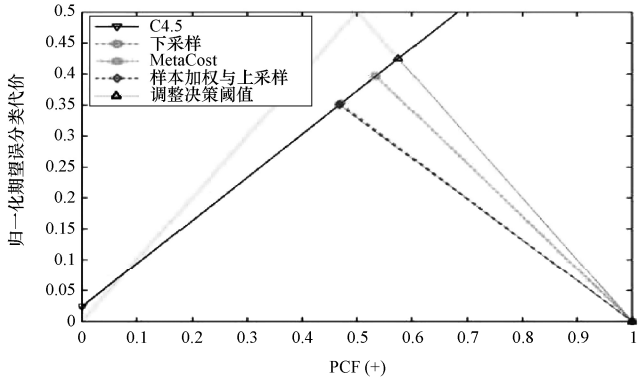


(a)

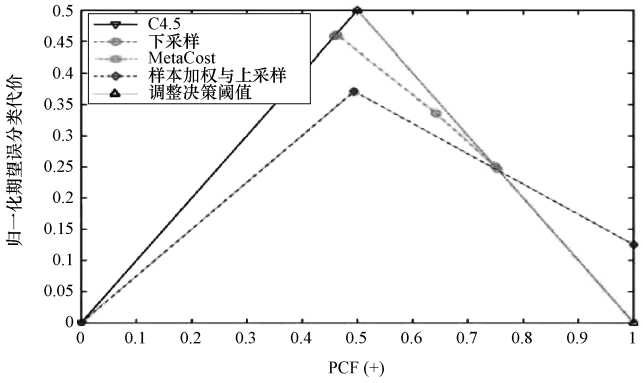


(b)

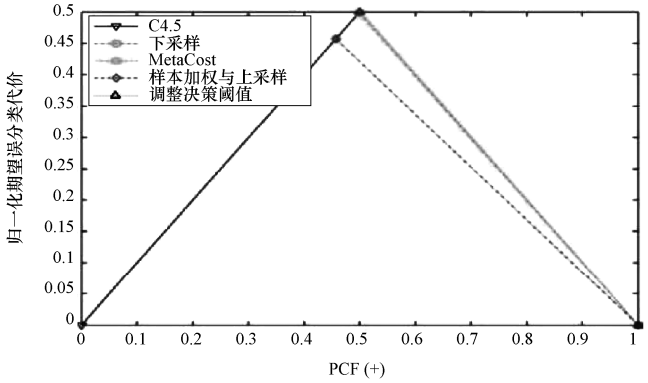
图 2-12　代价曲线



(c)

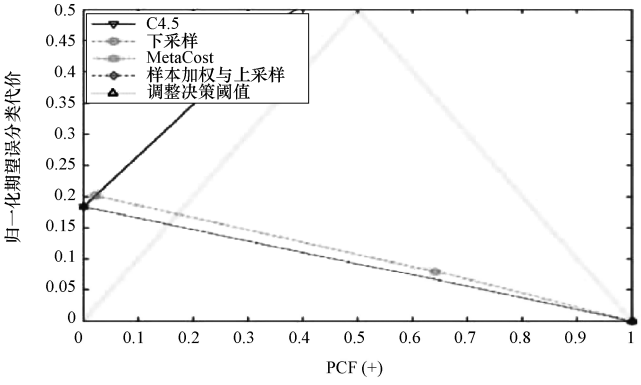


(d)

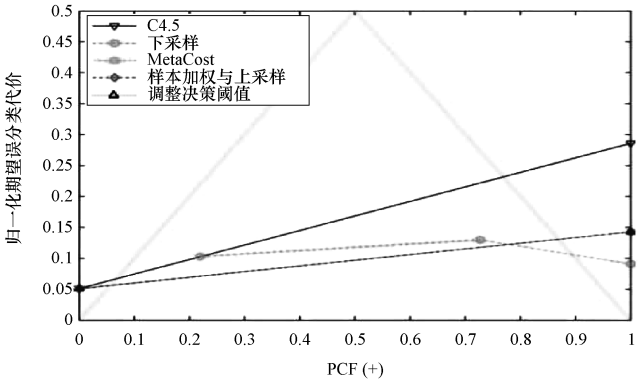


(e)

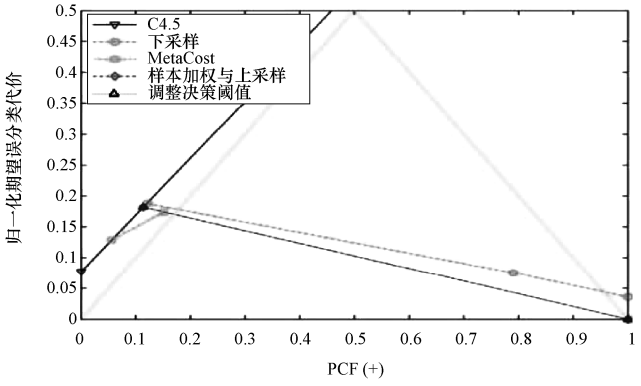
图 2-12　代价曲线（续）



(f)

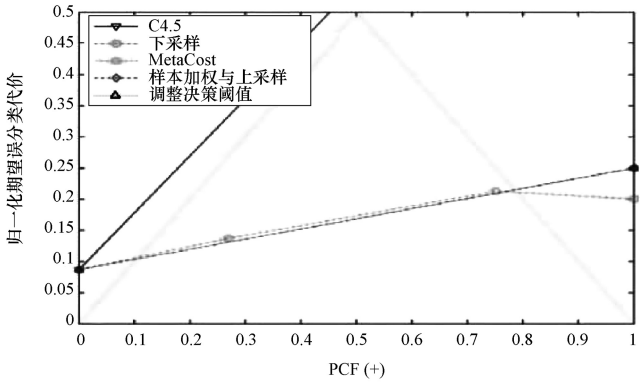


(g)

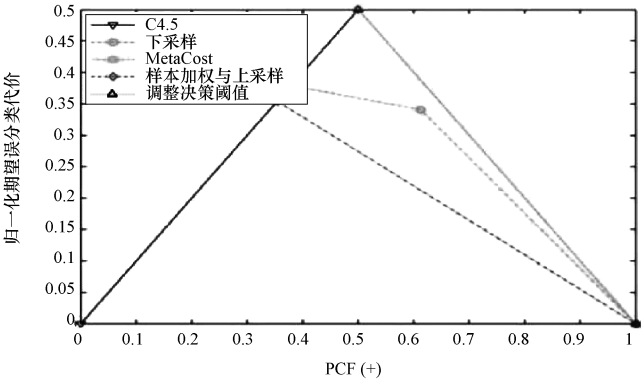


(h)

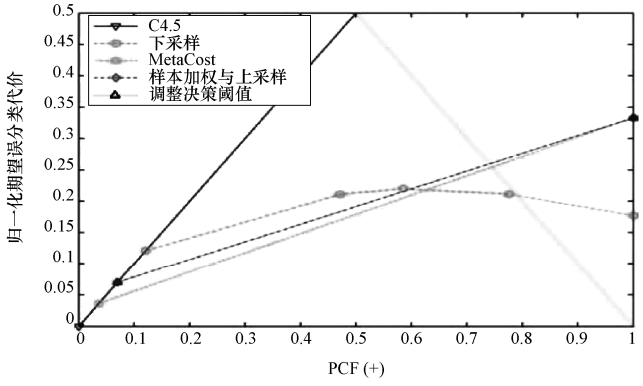
图 2-12　代价曲线（续）



(i)

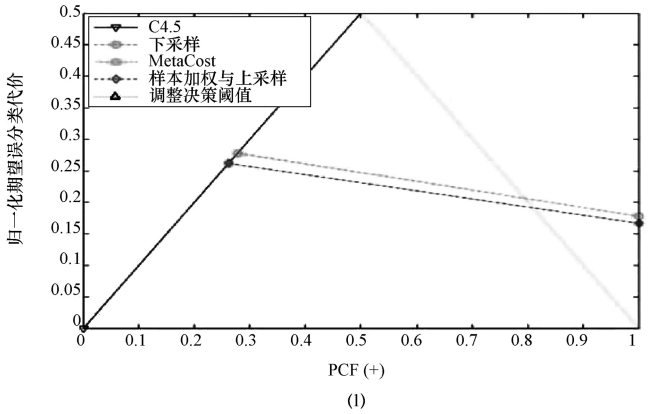


(j)

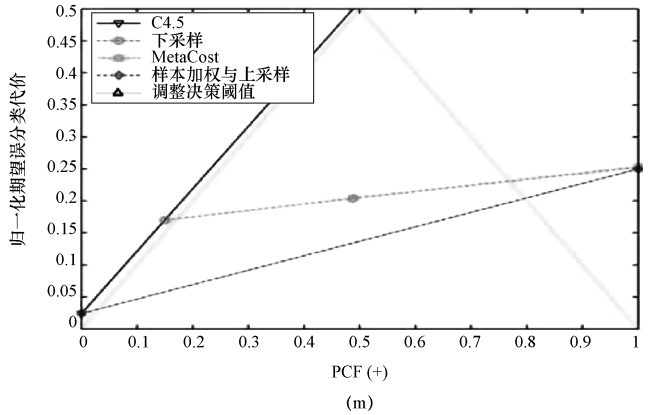


(k)

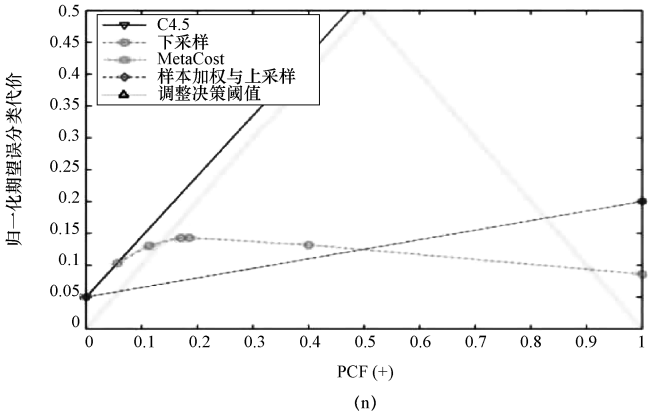
图 2-12　代价曲线（续）



(l)



(m)



(n)

图 2-12　代价曲线（续）

第 3 章

可操作行为规则挖掘

尽管组织行为预测模型可提供相当准确的组织行为预测知识，但却不能提供可被用户直接用来影响（抑制或鼓励）组织行为并因此获益的具体行动建议。这些行动建议又称为可操作知识，常常是用户切实需要的。尽管已有不少研究致力于其他类型的可操作知识的发现，挖掘影响组织行为的可操作规则（或可操作行为规则）这一重要问题尚未被识别、定义和研究。

本章建立了一类新的组织行为模式挖掘问题——可操作（组织）行为规则挖掘。3.1 节提出了可操作行为规则挖掘问题的形式化定义。3.2 节提出两个可操作行为规则挖掘算法 MABR-1 和 MABR-2。3.3 节提出可操作行为规则挖掘算法（模型）的经验验证方法及一个基准方法，并对 MABR-1 和 MABR-2 的有效性进行了验证。3.4 节对几个可操作行为规则挖掘的相关问题进行了深入讨论。

3.1 问题定义

本节通过一系列的形式化定义来阐明可操作行为规则挖掘问题。

组织行为的信息可用一张信息表来表示，也就是说，组织行为信息是一张特殊的信息表。文献[112]为信息表提供了一个严格并易于遵循的形式化定义——信息系统。因此，本节首先借鉴信息系统的定义为组织行为信息进行形式化定义。

定义 3-1 关于某组织的行为信息系统（Behavioral Information System）定义为一个 5 元组 $I=(O,o^*,A,D,\rho)$，其中 O 是对实体（组织）的观察的有限非空集，$o^*\in O$ 是下一个观察的投射，A 是属性的有限非空集，$D=U_{a\in A}D_a$（D_a 是属性 a 的值域），$\rho:O\times A\to D$ 是一个将每个观察和属性值的集合关联起来的函

数。A 可进一步分为两个子集，即 $A = A_{en} \cup A_{be}$，其中，A_{be} 是描述组织行为的行为属性的集合，A_{en} 是描述组织所处环境并对行为属性有影响的环境属性的集合。

除 o^* 外，每个关于环境和行为属性的观察均来自一个特定间隔的时间段。o^* 是一个基于最近观察的对将要到来特定时间段的观察的投射（假设该时间段内用户没有采取任何行动以影响组织行为）。例如，假定某恐怖组织近来频繁发动恐怖袭击，如果政府不采取任何有针对性的行动，那么在某时间段内（如半年），严峻形势将保持不变或改变很小。可操作行为规则的目标就是识别可改善下一个观察 o^* 的有益行动建议。在上例中，政府应该非常想知道采取何种行动才能减少恐怖袭击的频度并取得满意的效果。

不失一般性，假定所有属性都是类别属性。如果存在数值属性，则提前对其进行离散化处理。注意，行为属性并不限于指示某行为是否发生的二元属性，它也可以描述某行为的频度、烈度等。A_{be} 是一个涵盖实体（组织）所有相关行为的综合集。

例 3-1　考虑一个关于某组织的假想的行为信息系统 $I = (O, o^*, A, D, \rho)$，其中，$O = \{o^*, o_1, o_2, \cdots, o_{10}\}$，$A_{en} = \{e_1, e_2\}$，$A_{be} = \{b_1, b_2\}$，$D_{e_1} = D_{e_2} = \{0,1\}$，$D_{b_1} = D_{b_2} = \{0,1,2\}$，行为信息系统 I 中的 ρ 函数如表 3-1 所示（例如，$\rho(o_1, e_1) = 1$，$\rho(o_3, b_2) = 2$）。各属性的含义与取值包含在本书附录中。在下文中将以本例为基础构建其他例子。

表 3-1　行为信息系统 I 中的 ρ 函数

	ORSTPOLSUP	ORGCULTGR	DEMORGVIOLENCE	TRANSVIOLENCE
	e_1	e_2	b_1	b_2
o^*	1	1	2	2
o_1	1	1	2	2
o_2	1	0	2	1
o_3	1	0	1	2
o_4	1	0	1	2
o_5	0	1	1	1

（续表）

	ORSTPOLSUP	ORGCULTGR	DEMORGVIOLENCE	TRANSVIOLENCE
	e_1	e_2	b_1	b_2
o_6	0	1	2	1
o_7	0	1	2	1
o_8	0	0	0	0
o_9	0	0	0	1
o_{10}	0	0	1	0

定义 3-2 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。行动（Action）定义为一个三元组 $t=(a,v_f,v_t)$，其中，$a\in A_{en}$，$v_f=\rho(o^*,a)$，$v_t\in D_a$。如果 $v_f=v_t$，则称 t 为非行动（non-Action）。如果 a 的值从 v_f 变到 v_t，则称行动 $t=(a,v_f,v_t)$ 满足（Holds）。行动集（Action Set）S 又称 $|S|$-行动集，定义为行动的有限非空集，其中，对任何 $t_1,t_2\in S$，有 $t_1\cdot a\neq t_2\cdot a$。如果每个 $t\in S$ 都满足，则称行动集 S 满足。如果行动集 S 是行动集 S' 的真超集而且 S'/S 仅包含非行动，则称 S 是关于 S' 的一个一般行动集（General Action Set），而 S' 是关于 S 的一个具体行动集（Specific Action Set）。如果对每一个 $t\in S$ 有 $\rho(o,t.a)=t.v_t$，则称观察 o 支持（Supports）S。S 的支持度（Support）定义为

$$\sup(S)=|\{o\in O\mid o \text{ supports } S\}|$$

如果 $\sup(S)\geqslant$ minsup，则称 S 为关于一个用户指定的阈值——最小支持度（Minsup）的频繁行动集（Frequent Action Set）或者频繁$|S|$-行动集。如果行动集 S 是一个频繁行动集且其任一真超集均不是频繁行动集，则称 S 是最具体频繁行动集（Most Specific Frequent Action Set）。

例 3-2 $(e_1,1,0)$ 是一个行动。$(e_1,1,1)$ 是一个非行动。$(e_1,0,1)$ 不是一个行动。$\{(e_1,1,0)\}$ 是一个 1-行动集，$\{(e_1,1,0)(e_2,1,0)\}$ 是一个 2-行动集。$\{(e_1,0,1)\}$ 和 $\{(e_1,1,0)(e_1,0,1)\}$ 都不是行动集。$\{(e_1,1,0)\}$ 是关于 $\{(e_1,1,0)(e_2,1,1)\}$ 的一个一般行动集，而 $\{(e_1,1,0)(e_2,1,1)\}$ 是关于 $\{(e_1,1,0)\}$ 的一个具体行动集。如果属性 ORSTPOLSUP 的值从 1 变为 0，则行动 $(e_1,1,0)$ 满足。如果 $(e_1,1,0)$ 和 $(e_2,1,1)$ 均满足，则行动集 $\{(e_1,1,0)(e_2,1,1)\}$ 满足。$\sup(\{(e_1,1,0)\})=|\{o_5,o_6,o_7,o_8,o_9,o_{10}\}|=6$，

$\sup(\{(e_1,1,0),(e_2,1,1)\}) = |\{o_5,o_6,o_7\}|=3$。给定最小支持度为 2，则$\{(e_1,1,0)\}$是一个频繁 1-行动集，$\{(e_1,1,0)(e_2,1,1)\}$是一个频繁 2-行动集，同时也是一个最具体频繁行动集。给定最小支持度为 4，则$\{(e_1,1,0)\}$是一个频繁 1-行动集，同时也是一个最具体频繁行动集，但不是一个频繁 2-行动集。

定义 3-3　$I=(O,o^*,A,D,\rho)$是一个行为信息系统。效果（Effect）定义为一个三元组$e=(a,v_f,v_t)$，其中，$a\in A_{be}$，$v_f=\rho(o^*,a)$，$v_t\in D_a$。注意v_f与v_t可以相等。如果 a 的值从 v_f变为 v_t，则称效果 e=(a,v_f,v_t）发生（Takes Place）。效果—概率（Effect-Probability）定义为 ep=(e,p)，其中，e 是一个效果，而 $p\in[0,1]$。如果 e 以概率 p 发生，则称效果−概率 ep=(e,p)发生。

例 3-3　$(b_1,2,0)$是一个效果，而$(b_1,0,2)$不是一个效果。如果属性 DEMORGVIOLENCE 的值从 2 变到 0，则效果$(b_1,2,0)$发生。$((b_1,2,0),0.5)$是一个效果—概率。如果$(b_1,2,0)$以概率 0.5 发生，则效果—概率$((b_1,2,0),0.5)$发生。

定义 3-4　$I=(O,o^*,A,D,\rho)$是一个行为信息系统。原子可操作行为规则（Atomic Actionable Behavioral Rule）定义为$ar=(S,e)$，其中，S 是一个最具体频繁行动集，e 是一个效果。如果对任一$t\in S$有$\rho(o,t.a)=t.v_t$，且$\rho(o,e.a)=e.v_t$，则称 o 支持 ar。ar 的置信度（confidence）定义为

$$\mathrm{conf}(\mathrm{ar})=|\{o\in O\mid o\,\mathrm{supports\,ar}\}|/\sup(S)$$

原子可操作行为规则的置信度可被认为是对当行动集满足时效果发生的可能性的概率估计。

例 3-4　$(\{(e_1,1,0)\},(b_1,2,0))$是一个原子可操作行为规则。$\mathrm{conf}((\{(e_1,1,0)\},(b_1,2,0)))=|\{o_8,o_9\}|/6=1/3$。

定义 3-5　$I=(O,o^*,A,D,\rho)$是一个行为信息系统。候选可操作行为规则（Candidate Actionable Behavioral Rule）定义为$cr=(S,C)$，其中，C 是效果−概率的一个有限非空集，对任一$ep\in C$，$(S,\mathrm{ep.e})$是一个置信度为 ep.p 的原子可操作行为规则，$|C|=\sum_{a\in A_{be}}|D_a|$。候选可操作行为规则 cr=(S,C)的支持度定义为

$\sup(\mathrm{cr}) = \sup(S)$。

例 3-5 $\left(\{(e_1,1,0)\},\begin{Bmatrix}((b_1,2,0),1/3),((b_1,2,1),1/3),((b_1,2,2),1/3),\\((b_2,2,0),1/3),((b_2,2,1),2/3),((b_2,2,2),0)\end{Bmatrix}\right)$ 是一个候选可操作行为规则，表示为 r。其含义为如果行动集 $\{(e_1,1,0)\}$ 满足，则效果—概率 $((b_1,2,0),1/3)$，$((b_1,2,1),1/3)$，$((b_1,2,2),1/3)$，$((b_2,2,0),1/3)$，$((b_2,2,1),2/3)$，和 $((b_2,2,2),0)$ 将发生。$\sup(r) = \sup(\{(e_1,1,0)\}) = 6$。

改变环境属性的值和行为属性的值都会为用户带来收益（正效用）或损失（负效用）。换句话说，行动或效果会或正或负。很明显，如果环境或行为属性值没有变化，则相应的效用为 0。

定义 3-6 $I = (O, o^*, A, D, \rho)$ 是一个行为信息系统。候选可操作行为规则 $cr = (S, C)$ 的期望效用（Expected Utility）定义为

$$\mathrm{util}(\mathrm{cr}) = \sum_{t \in S} \mathrm{util}(t) + \sum_{\mathrm{ep} \in C} \mathrm{util}(\mathrm{ep.e}) \cdot \mathrm{ep.p}$$

其中，$\mathrm{util}(t)$ 和 $\mathrm{util}(\mathrm{ep.e})$ 分别表示行动 t 和效果 ep.e 的效用。

例 3-6 假定 $(e_1,1,0)$，$(e_2,1,0)$，$(b_1,2,0)$，$(b_1,2,1)$，$(b_2,2,0)$ 和 $(b_2,2,1)$ 的效用分别是−1，−2，5，2，3 和 1。则 $\left(\{(e_1,1,0)\}\begin{Bmatrix}((b_1,2,0),1/3),((b_1,2,1),1/3),((b_1,2,2),1/3),\\((b_2,2,0),1/3),((b_2,2,1),2/3),((b_2,2,2),0)\end{Bmatrix}\right)$ 的期望效用是 $u((e_1,1,0)) + u((b_1,2,0)) \times 1/3 + u((b_1,2,1)) \times 1/3 + u((b_2,2,0)) \times 1/3 + u((b_2,2,1)) \times 2/3 = 3$。

定义 3-7 $I = (O, o^*, A, D, \rho)$ 是一个行为信息系统。在候选可操作行为规则集 CR 上的二元等价关系～定义为 $\mathrm{cr}_1, \mathrm{cr}_2 \in \mathrm{CR}$，对任一 $a \in \mathrm{cr}_1.\mathrm{S}$，如果 $a.v_f \neq a.v_t$，则 $a \in \mathrm{cr}_2.\mathrm{S}$，且对任一 $b \in \mathrm{cr}_2.\mathrm{S}$，如果 $b.v_f \neq b.v_t$，则 $b \in \mathrm{cr}_1.\mathrm{S}$，则 $\mathrm{cr}_1 \sim \mathrm{cr}_2$。关于在 CR 上的关系～的一个最大等价类 L 的合并可操作行为规则（Consolidated Actionable Behavioral Rule）定义为 $r = (S, C)$，其中，$S = \{a \in \bigcap_{\mathrm{cr} \in L} \mathrm{cr}.S \mid a$不是一个非行动$\}$，$C$ 是一个效果—概率的有限非空集，$|C| = \sum_{a \in A_{\mathrm{be}}} |D_a|$，对任一 $ep \in C$，有 $\mathrm{ep.p} = \sum_{\mathrm{cr} \in L} \mathrm{conf}((\mathrm{cr.S}, \mathrm{ep.e})) \cdot \sup(\mathrm{cr.S}) / \sum_{l \in L} \sup(l.S)$。合并可操作行为规则 $r = (S, C)$ 的期望效用（expected utility）定义为

$$\text{util}(r)=\sum_{t\in S}\text{util}(t)+\sum_{ep\in C}\text{util}(ep.e)\cdot ep.p$$

其中，util(t) 和 util(ep.e) 分别表示行动 t 和效果 $ep.e$ 的效用。如果 util(r)≥minutil，则称合并可操作行为规则 r 为关于用户指定阈值最小效用（Minutil）的有趣可操作行为规则（Interesting Actionable Behavioral Rule）。

合并可操作行为规则 r 的含义是，如果行动集 $r.S$ 满足，则对任一 $\text{ep}\in r.C$，效果 ep.e 将以概率 ep.p 发生。在实际应用中，仅当一条规则的期望效用超出某个阈值时，用户才会认为其是“有趣”的。

例 3-7　$\left(\{(e_1,1,0)\},\left\{\begin{array}{c}((b_1,2,0),1/3),((b_1,2,1),1/3),((b_1,2,2),1/3),\\((b_2,2,0),1/3),((b_2,2,1),2/3),((b_2,2,2),0)\end{array}\right\}\right)$

等价于 $\left(\left\{\begin{array}{c}(e_1,1,0),\\e_2,1,1\end{array}\right\},\left\{\begin{array}{c}((b_1,2,0),0),((b_1,2,1),1/3),((b_1,2,2),2/3),\\((b_2,2,0),0),((b_2,2,1),1),((b_2,2,2),0)\end{array}\right\}\right)$

但不等价于 $\left(\left\{\begin{array}{c}(e_1,1,0),\\(e_2,1,0)\end{array}\right\},\left\{\begin{array}{c}((b_1,2,0),2/3),((b_1,2,1),1/3),((b_1,2,2),0),\\((b_2,2,0),2/3),((b_2,2,1),1/3),((b_2,2,2),0)\end{array}\right\}\right)$

$\left(\{(e_1,1,0)\},\left\{\begin{array}{c}((b_1,2,0),2/9),((b_1,2,1),1/3),((b_1,2,2),4/9),\\((b_2,2,0),2/9),((b_2,2,1),7/9),((b_2,2,2),0)\end{array}\right\}\right)$

是一个关于在 $\left\{\begin{array}{l}\left(\{(e_1,1,0)\},\left\{\begin{array}{c}((b_1,2,0),1/3),((b_1,2,1),1/3),((b_1,2,2),1/3),\\((b_2,2,0),1/3),((b_2,2,1),2/3),((b_2,2,2),0)\end{array}\right\}\right),\\\left(\left\{\begin{array}{c}(e_1,1,0),\\(e_2,1,1)\end{array}\right\},\left\{\begin{array}{c}((b_1,2,0),0),((b_1,2,1),1/3),((b_1,2,2),2/3),\\((b_2,2,0),0),((b_2,2,1),1),((b_2,2,2),0)\end{array}\right\}\right)\end{array}\right\}$ 上的关系～的唯一最大等价类的合并可操作规则。该合并可操作规则的期望效用为 20/9。

以上的合并可操作规则 r 实际上提出了以下观点：“如果将 ORSTPOLSUP 从级别 1 变为 0，则 DOMORGVIOLENCE 将从级别 2 以 2/9 的概率变为 0，或以 1/3 的概率变为 1，或以 4/9 的概率保持不变；TRANSVIOLTARG 将从级别 2 以 2/9 的概率变为 0，或以 7/9 的概率变为 1。”

可操作行为规则的定义已经完成。挖掘可操作行为规则就是从一个行为信息系统中挖掘所有的有趣、可操作行为规则。阈值 minsup 用来排除规则发现的偶然性，而阈值 minutil 用来确保规则对用户足够有益。

注意，在可操作行为规则挖掘中，最小支持度限制仅用来确保统计显著性，而在关联规则挖掘[112, 113]中（如购物篮分析等），该限制也被用于一些实际情况[113]。也就是说，如果一条规则没有足够的支持则不值得被考虑。本书仅考虑最具体行动集，因为关于一个具体行动集的原子规则的置信度的估计比关于一个一般行动集的更可信。等价的候选规则根据其支持度合并，因为它们提供的行动建议相同，而效果概率及期望效用不同。

3.2　挖掘算法

为挖掘可操作行为规则，本节提出了两个算法：MABR-1（Mining Actionable Behavioral Rules-1）和 MABR-2。本节将详细描述这两个算法。

3.2.1　MABR-1 算法

MABR-1（见下文框内）分为两个阶段，分别是最具体行动集产生阶段和有趣可操作行为规则产生阶段。

第一个阶段类似于用于关联规则挖掘[112]的 Apriori 算法中的频繁项集的产生过程。该阶段的核心思想是频繁行动集的下闭包特征。也就是说，如果一个行动集的支持度超过 *minsup* 阈值，则其所有子集的支持度必超过 *minsup* 阈值。这一特征用来有效减少需要被核实的潜在的频繁行动集的数量。

从频繁 1−行动集开始（行 1，调用函数 Select），算法迭代发现频繁 2−行动集、频繁 3−行动集等，直到没有频繁 k−行动集产生（行 3～8）。在每次迭代中，频繁 k−行动集被用来产生潜在的频繁$(k+1)$−行动集（行 4，调用函数 Generate），然后通过计算支持度，检查是否是频繁$(k+1)$−行动集。对任一频繁 k−行动集，如果其为关于任一频繁$(k+1)$−行动集的一般行动集，则将其删除（行 6、7）。阶段 1 输出所有最具体频繁行动集的集合。

函数 Generate 的输入是频繁 k−行动集的集合，输出是潜在的频繁$(k+1)$−行动集的集合（行 18～30）。首先，对任何频繁 k−行动集 S_1 和 S_2，如果它们中有且仅有一个元素不同，则（S_1，S_2）合并为一个潜在的频繁$(k+1)$−行动集（行

19～21)。根据下闭包特征，如果一个潜在的频繁(k+1)–行动集的任一 k–子集不是频繁行动集，则将其删除（行 22～29）。

输入：$I=(O,o^*,A,D,\rho)$，所有可能的行动和效果的效用，minsup，minutil

输出：有趣的可操作行为规则及其期望效用

// 第 1 阶段：产生最具体频繁行动集

1. $F_1 \leftarrow$ Select ({1−action sets})
2. $k \leftarrow 1$
3. **while** $F_k \neq \varnothing$
4. $F_{k+1} \leftarrow \text{Generate}(F_k)$
5. $F_{k+1} \leftarrow \text{Select}(F_{k+1})$
6. **for each** $S \in F_{k+1}$
7. 从 F_k 中删除关于 S 的一般行动集
8. $k \leftarrow k+1$
9. $F \leftarrow \bigcup_{i=1}^{k-1} F_i$

// 第 2 阶段：产生有趣的可操作行为规则

10. **for each** $S \in F$
11. cr ← 以 S 为前件的候选可操作行为规则
12. $CR \leftarrow CR \cup \{cr\}$
13. **for each** 关系~在 CR 上的最大等价类 LE
14. r ← LE 中的合并可操作行为规则
15. **if** $util(r) \geqslant minutil$
16. $R \leftarrow R \cup \{(r, util(r))\}$
17. **return** R
18. **Function** Generate(F: set of action sets)
19. **for each** $\{S_1,S_2\} \subset F$,满足 $|S_1 \setminus S_2| = 1$
20. **if** $S_1 \cup S_2$ 是一个行动集

```
21.            C ← C ∪ {S1 ∪ S2}
22.  for each   c ∈ C
23.        flag ← 1
24.        for each c′ ⊂ c，满足 |c \ c′| = 1
25.            if  c′ ∉ F
26.                flag ← 0
27.                break
28.        if  flag = 0
29.            C ← C \ {c}
30.  return C

31.  Function Select(C: set of action sets)
32.  for each   o ∈ O
33.        Co ← {c ∈ C | o supports c}
34.        for each   c ∈ Co
35.            c.sup ← c.sup + 1
36.  for each   c ∈ C
37.        if  c.sup < minsup
38.            C ← C \ {c}
39.  return C
```

函数 Select 的输入是潜在的频繁(k+1)–行动集的集合，输出是频繁(k+1)–行动集的集合（行 31～39）。仅需扫描一次数据集就可计算所有潜在的频繁(k+1)–行动集的支持度（行 32～35）。支持度低于 minsup 的行动集被删除（行 36～38）。

第二个阶段基于第一个阶段产生的最具体频繁行动集生成有趣可操作行为规则集（行 10～17）。首先，以每个最具体频繁行动集为前件，构建一条候选可操作行为规则（行 10～12）。然后，基于可操作行为规则集合上关系～的每个最大等价类构建一条合并可操作行为规则，若其期望效用大于 minutil，则将其包括在最终输出中（行 13～16）。

例 3-8 基于之前的例子，假定 minsup 和 minutil 被分别设为 3 和 2，MABR-1 的运行可粗略地描述如下：①频繁 1−行动集的集合 $\{\{(e_1,1,1)\},\{(e_1,1,0)\},\{(e_2,1,1)\},\{(e_2,1,0)\}\}$ 生成；②潜在的频繁 2−行动集的集合 $\left\{\left\{\begin{matrix}(e_1,1,1),\\(e_2,1,1)\end{matrix}\right\},\left\{\begin{matrix}(e_1,1,1),\\(e_2,1,0)\end{matrix}\right\},\left\{\begin{matrix}(e_1,1,0),\\(e_2,1,1)\end{matrix}\right\},\left\{\begin{matrix}(e_1,1,0),\\(e_2,1,0)\end{matrix}\right\}\right\}$ 生成；③频繁 2−行动集的集合 $\{\{(e_1,1,1),(e_2,1,0)\},\{(e_1,1,0),(e_2,1,1)\},\{(e_1,1,0),(e_2,1,0)\}\}$ 生成；④最具体频繁行动集的集合 $\{\{(e_1,1,1),(e_2,1,0)\},\{(e_1,1,0),(e_2,1,1)\},\{(e_1,1,0),(e_2,1,0)\}\}$ 生成；⑤候选可操作行为规则集的集合

$$\left\{\begin{matrix}\left(\left\{\begin{matrix}(e_1,1,1),\\(e_2,1,0)\end{matrix}\right\},\left\{\begin{matrix}((b_1,2,0),0),((b_1,2,1),2/3),((b_1,2,2),1/3),\\((b_2,2,0),0),((b_2,2,1),1/3),((b_2,2,2),2/3)\end{matrix}\right\}\right),\\\left(\left\{\begin{matrix}(e_1,1,0),\\(e_2,1,1)\end{matrix}\right\},\left\{\begin{matrix}((b_1,2,0),0),((b_1,2,1),1/3),((b_1,2,2),2/3),\\((b_2,2,0),0),((b_2,2,1),1),((b_2,2,2),0)\end{matrix}\right\}\right),\\\left(\left\{\begin{matrix}(e_1,1,0),\\(e_2,1,0)\end{matrix}\right\},\left\{\begin{matrix}((b_1,2,0),2/3),((b_1,2,1),1/3),((b_1,2,2),0),\\((b_2,2,0),2/3),((b_2,2,1),1/3),((b_2,2,2),0)\end{matrix}\right\}\right)\end{matrix}\right\}$$

生成；⑥有趣可操作行为规则及其期望效用的集合

$$\left\{\begin{matrix}\left(\left(\{(e_1,1,0)\},\left\{\begin{matrix}((b_1,2,0),0),((b_1,2,1),1/3),((b_1,2,2),2/3),\\((b_2,2,0),0),((b_2,2,1),1),((b_2,2,2),0)\end{matrix}\right\}\right),3\right),\\\left(\left(\left\{\begin{matrix}(e_1,1,0),\\(e_2,1,0)\end{matrix}\right\},\left\{\begin{matrix}((b_1,2,0),2/3),((b_1,2,1),1/3),((b_1,2,2),0),\\((b_2,2,0),2/3),((b_2,2,1),1/3),((b_2,2,2),0)\end{matrix}\right\}\right),\frac{10}{3}\right)\end{matrix}\right\}$$

生成并作为最终输出返回。

3.2.2 MABR-2 算法

MABR-1 的第一个阶段需要迭代发现不断增加频繁行动集，这成为整个算

法的效率瓶颈。在每次迭代中，首先通过合并上次迭代生成的频繁行动集以生成潜在的频繁行动集，然后扫描行为信息系统以计算本次迭代中的潜在的频繁行动集的支持度。这一过程因不断地扫描数据库而特别耗时。假如行为信息系统的属性比较多而且（或者）属性的值域较大，支持度阈值较小，则每次迭代中的潜在的频繁行动集的数目就可能是巨大的，MABR-1 就会很耗时。

相比较 MABR-1，MABR-2（见下文框内）有良好的可伸缩性和效率。它避免了潜在的可操作行为规则的产生−检测步骤，并使用一个 FA-tree 数据结构以显著地减少计算代价。基于 FP-tree[114]的定义，我们定义了 FA-tree，conditional subtree 和 conditional FA-tree。

定义 3-8　$I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。FA-tree（Frequent Action Set Tree）定义为一个树结构：

① 其包含一个根，一个根的 1−行动集−前缀的子树的集合和一个列表 header。

② 1−行动集−前缀子树的每个节点有四个域：1−action−set，count，next 和 parent。域 1−action−set 载有一个 1−行动集。域 count 记录 O 中支持从根到该节点路径上的所有节点载有的 1−行动集的合集的观察的数目。域 next 链接 FA-tree 中 1−action−set 域与该节点的 1−action−set 域相同的下一个节点。域 parent 链接该节点的父节点。

③ 列表 header 中的每个元素有两个域：1−action−set 和 first。域 1−action−set 载有一个 1−行动集。域 first 指向 FA-tree 中其 1−action−set 域与该元素的 1−action−set 域相同的下一个节点。

定义 3-9　$I=(O,o^*,A,D,\rho)$ 是一个行为信息系统，T 是一棵关于 I 的 FA-tree。*T.header* 中元素 α 的 conditional subtree 定义为 T 的一棵包含其 1−action−set 域与 α 的 1−action−set 域相同的所有节点与从根节点到这些节点的路径上所有节点的子树。conditional FA-tree 定义为一棵以 α 存在为条件的 FA-tree，其可基于 α 的 conditional subtree 构建。

MABR-2 比 MABR-1 更高效且可伸缩性更好的主要原因有两个：一是 MABR-2 仅须扫描很少几次行为信息系统；二是 FA-tree 比行为信息系统要小得多。

输入：$I=(O,o^*,A,D,\rho)$，所有可能的行动和效果的效用，minsup，minutil.

输出：有趣的可操作行为规则及其期望效用

1. F←{频繁 1−行动集}
// 构造 FA-tree
2. T←一个 FA-tree 的根节点（无孩子节点）
3. T.header←以支持度降序排列的二元组（频繁 1−行动集，null）列表
4. **for each** $o \in O$
5. L←空列表
6. **for each** $a \in A_{en}$
7. **if** $S=\{(a,p(o^*,a),p(o,a))\} \in F$
8. L.Add(S)
9. 根据 T.header 的次序对 L 排序
10. Insert(L,T)
// 产生最具体频繁行动集
11. Construct(T, null)
12. **for each** i from 1 to $|A_{en}|$
13. **if** set of (i+1)−action sets $\neq \varnothing$
14. 从 F 中删除关于任何(i+1)−行动集的具体 i−行动集
// 产生有趣的可操作行为规则
15. **for each** $S \in F$
16. cr ← 以 S 为前件的候选可操作行为规则
17. CR ← CR ⋃ {cr}
18. **for each** 关系~在 CR 上的最大等价类 LE
19. r←LE 中的合并可操作行为规则
20. **if** $util(r) \geqslant$ minutil

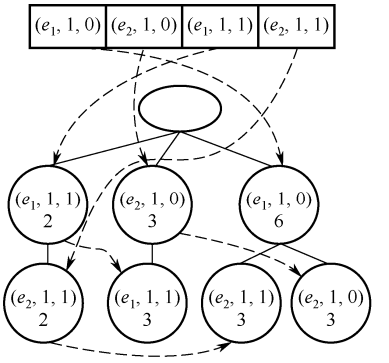
```
21.            R ← R ⋃ {(r, util(r))}
22.    return R
23.    Procedure Insert(L: 1−行动集列表, P: PA-tree 节点)
24.    if   P 没有这样的孩子节点 N, 满足 N.1−action−set = L[1]
25.        i ← j，满足 header[j].1−action−set = L[1]
26.        N ← FA−tree node (L[1], 1, T.header[i].first, P)
27.        T.header[i].first ←N
28.    else
29.        N.count←N.count+1
30.    L.Delete(1)
31.    if L 非空
32.        Insert(L, N)
33.    Procedure Construct(P: FA-tree, A:  行动集)
34.    if P 有一条单一路径
35.        for each P 的除根外的所有节点的 1−action−set 域的组合 c
36.            F ← F ⋃ {c ⋃ A}
37.    else
38.        for each P.header 中的元素 I （自尾向首）
39.            A←A ⋃ I.1−action−set
40.            CB←I 的条件 subtree
41.            CT←CB 的条件 FA-tree
42.            if CT≠null
43.                Construct(CT, A)
```

MABR-2 采用了 FP-growth 算法[114]的一个变体，总共只需要扫描 3 次行为信息系统。第一次扫描收集所有频繁 1−行动集（行 1）；第二次扫描构建 FA-tree；第三次扫描生成候选可操作行为规则（行 15～17）。不管是基于 FA-tree 生成最具体行动集的集合（行 11～14），还是基于候选规则生成合并规则（行 18～21），都不需要扫描行为信息系统。生成有趣的可操作行为规则的步骤（行 15～21）

与 MABR-1 是不同的。递归过程 Insert（行 23～32）将某观察支持的排序频繁 1−行动集插入 FA-tree。递归过程 Construct（行 33～43）构建所有频繁 k−行动集（$k>1$）。

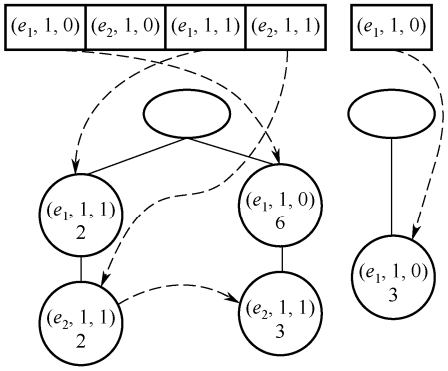
一棵 FA-tree 的大小（节点数）最多为 $\sum_{o\in O}|L(o)|+1$，高度最多为 $\mathrm{MAX}_{o\in O}|L(o)|+1$，其中 $L(o)$表示观察 o 支持的频繁行动集的列表（由行 6～8 得出）。这意味着一棵 FA-tree 的规模小于相应的行为信息系统的规模。因为不同观察支持的不同的频繁行动集列表可能含有相同项，所以一棵 FA-tree 的规模通常远小于相应的行为信息系统的规模。

例 3-9 基于之前的例子，假定 minsup 和 minutil 被分别设为 3 和 2，MABR-2 的运行可粗略地描述如下：①频繁 1−行动集 $\{(e_1,1,0)\},\{(e_2,1,0)\},\{(e_1,1,1)\}$ 和 $\{(e_2,1,1)\}$ 生成；②FA−tree T 生成（图 3-1（a）），简单起见，图 3-1 中符号 (a,v_f,v_t) 表示行动集 $\{(a,v_f,v_t)\}$，如 $(e_1,1,0)$ 表示行动集 $\{(e_1,1,0)\}$；③关于 T.header[4]的 conditional subtree 和 conditional FA-tree（图 3-1（b））与频繁行动集 $\{(e_2,1,1),(e_1,1,0)\}$ 生成；④关于 T.header[3]的 conditional subtree 和 conditional FA-tree（图 3-1（c））与频繁行动集 $\{(e_1,1,1),(e_2,1,0)\}$ 生成；⑤关于 T.header[2] 的 conditional subtree 和 conditional FA-tree（图 3-1（d））与频繁行动集 $\{(e_2,1,0),(e_1,1,0)\}$ 生成；⑥最具体频繁行动集的集合生成。其余步骤同 MABR-1 的运行实例的最后两步。

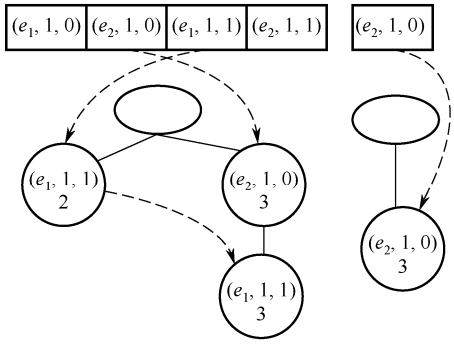


(a) FA-tree T

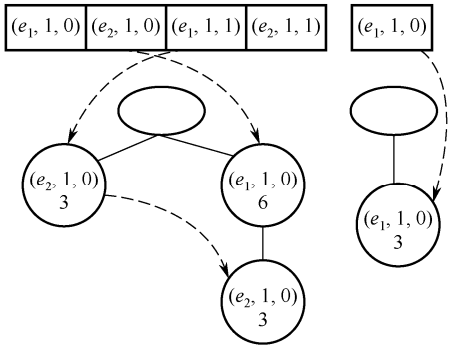
图 3-1 MABR-2 运行实例



(b) T.header[4]

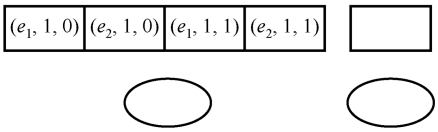


(c) T.header[3]



(d) T.header[2]

图 3-1　MABR-2 运行实例（续）



（e）T.header[1]的条件子树与条件FA-tress

图 3-1　MABR-2 运行实例（续）

3.3　模型验证

本节验证了提出的可操作规则挖掘方法，展示了一个挖掘出的规则的有趣例子，并且比较了 MABR-1 和 MABR-2 的运行时间。

1. 实验设置

实验数据集采用 MAROB 数据集。挖掘算法采用 java 实现。台式计算机装备 2.4GHz 奔腾 4CPU、2GB 主存和 Windows XP 操作系统。

从MAROB中抽取三个组织的数据作为三个行为信息系统，选取DOMORG VIOLENCE（对境内国内目标实施恐怖袭击活动的程度），TRANSVIOLTARG（对境内国际目标实施恐怖袭击活动的程度）与 TRANSVIOLOC（对境外目标实施恐怖袭击活动的程度）作为行为属性，选取多个与行为属性关系紧密的环境属性。所有行动和效果的效益值由领域专家指定并规格化到区间[−1，1]。

2. 基准方法

$I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。对任一行动集 S，相应的效果—概率的集合是相同的，可表示为 C，且对任意 $a\in A_{be}$ 和 $v\in D_a$，存在一个效果—概率 $\mathrm{ep}\in C$，$\mathrm{ep.e}=(a,\rho(o^*,a),v),\mathrm{ep.p}=|\{o\in O\mid\rho(o,a)=v\}|/|O|$。

基准方法采用如下策略计算任一行动集 S 的期望效用：

$$\mathrm{util}(S)=\sum_{t\in S}\mathrm{util}(t)+\sum_{\mathrm{ep}\in C}\mathrm{util}(\mathrm{ep.e})\cdot\mathrm{ep.p} \tag{3-1}$$

3. 评价指标

境内有恐怖组织的政府每年都可能采取行动抑制恐怖行为，而可能的行动和效果的信息在 MAROB 数据集都有记录。具体来说，当年采取的行动的实际

效用可以根据前一年和当年的观察计算得到。另外，算法 MABRs 也会给出以实际采取的行动集为前件的可操作行为规则及其期望效用。这样，就可以简单、直接地比较期望效用与实际效用，以评价算法 MABRs 的有效性。显然，差别越小则算法 MABRs 越有效。

假设第 i 年政府采用的实际行动集为 S_i，S_i 的实际效用为 SU_i。若本项目提出的算法的输出中存在一条其前件等于 S_i 的可操作行为规则 r，则算法估计 S_i 的期望效用 $\mathrm{EU}_i=\mathrm{util}(r)$。基准方法对 S_i 的期望效用估计 EU_i 可直接根据式（3-1）计算得出。那么，实际效用与本项目提出算法或基准方法估计的期望效用的差 $D_i=\mathrm{SU}_i-\mathrm{EU}_i$，绝对差 $\mathrm{AD}_i=|\mathrm{SU}_i-\mathrm{EU}_i|$。

实际效用与算法 MABRs 或基准方法估计的期望效用的差的置信区间是

$$(M-t_{n-1,a/2}\mathrm{SD}/\sqrt{n},M+t_{n-1,a/2}\mathrm{SD}/\sqrt{n})$$

其中，M，SD 和 $t_{n-1,a/2}$ 分别表示平均差、标准差与自由度为 n 的 t 分布值。考虑本领域的性能要求，a 取 0.05。如果 0 位于该区间内，那么算法 MABRs 或基准方法与实际没有显著性差异。

另外，平均绝对误差也能衡量算法 MABRs 与基准方法的性能，即平均绝对差越小则算法越好。

4. 实验结果

表 3-2 展现了 minsup 设为 5 时基于三个子 MAROB 数据集的实验结果，包括实际效用、实际效用与基准方法所得期望效用的差与绝对差，实际效用与算法 MABRs 所得期望效用的差与绝对差，差的置信区间、均值与标准差，绝对差的均值与标准差。

表 3-2　实验结果

行动集	实际效用	效用差			
		基准方法		MABRs	
		实际值	绝对值	实际值	绝对值
1	−0.006	0.091		−0.085	
2	−0.246	−0.184		0.123	

（续表）

行动集	实际效用	效用差			
		基准方法		MABRs	
		实际值	绝对值	实际值	绝对值
3	0.194	−0.298		−0.119	
4	0.072	−0.056		−0.064	
5	−0.008	−0.107		0.080	
6	−0.012	−0.103		0.071	
7	0.000	−0.115		0.057	
8	−0.003	−0.112		0.093	
9	−0.004	−0.111		0.118	
10	0.064	−0.179		0.000	
11	−0.004	−0.111		0.041	
12	−0.016	−0.099		0.000	
13	−0.006	−0.109		−0.053	
14	−0.003	−0.112		0.000	
15	0.154	−0.269		−0.147	
16	−0.017	−0.098		−0.041	
17	−0.016	−0.099		−0.093	
18	−0.004	−0.111		−0.041	
19	−0.086	0.175		0.012	
20	−0.011	0.015		0.012	
21	−0.011	0.015		0.012	
22	−0.006	0.015		0.015	
23	0.077	−0.062		−0.024	
24	−0.084	+0.175		0.003	
25	0.077	−0.062		−0.032	
26	0.071	−0.062		−0.024	
27	−0.004	0.098		−0.089	
28	−0.120	0.213		0.089	
29	−0.011	−0.027		0.062	
30	−0.010	−0.027		0.047	
31	0.117	−0.142		−0.024	
均值		**−0.060**	**0.092**	**0.000**	**0.054**
标准差		**0.118**	**0.070**	**0.069**	**0.041**
置信区间	上界	**−0.103**		**−0.025**	
	下界	**−0.017**		**0.025**	

下面展示一个算法 MABRs 基于某组织子数据集挖掘出的可操作行为规则的例子。当 minsup、minutil 和 o^*分别被设为 8、0.05 和 2003 年的观察时，算法发现了三条有趣的可操作行为规则：

$$\left(\left(\{(e_2,1,0)\},\left\{\begin{array}{c}((b_1,2,0),0.09),((b_1,2,1),0),((b_1,2,2),0.91),\\((b_1,2,3),0),((b_1,2,4),0),((b_1,2,5),0),\\((b_2,2,0),0.04),((b_2,2,1),0),((b_2,2,2),0.85),\\((b_2,2,3),0.07),((b_2,2,4),0.04),((b_2,2,5),0),\\((b_3,5,0),0.33),((b_3,5,1),0),((b_3,5,2),0.07),\\((b_3,5,3),0.07),((b_3,5,4),0.13),((b_3,5,5),0.4)\end{array}\right\}\right),0.07\right)$$

$$\left(\left(\left\{\begin{array}{l}(e_2,1,0),\\(e_3,0,2)\end{array}\right\},\left\{\begin{array}{c}((b_1,2,0),0.3),((b_1,2,1),0),((b_1,2,2),0.7),\\((b_1,2,3),0),((b_1,2,4),0),((b_1,2,5),0),\\((b_2,2,0),0.1),((b_2,2,1),0),((b_2,2,2),0.6),\\((b_2,2,3),0.2),((b_2,2,4),0.1),((b_2,2,5),0),\\((b_3,5,0),0.6),((b_3,5,1),0),((b_3,5,2),0),\\((b_3,5,3),0),((b_3,5,4),0),((b_3,5,5),0.4)\end{array}\right\}\right),0.12\right)$$

$$\left(\left(\{(e_3,2,0)\},\left\{\begin{array}{c}((b_1,2,0),0.26),((b_1,2,1),0),((b_1,2,2),0.74),\\((b_1,2,3),0),((b_1,2,4),0),((b_1,2,5),0),\\((b_2,2,0),0.11),((b_2,2,1),0),((b_2,2,2),0.57),\\((b_2,2,3),0.21),((b_2,2,4),0.11),((b_2,2,5),0),\\((b_3,5,0),0.58),((b_3,5,1),0),((b_3,5,2),0),\\((b_3,5,3),0),((b_3,5,4),0),((b_3,5,5),0.42)\end{array}\right\}\right),0.13\right)$$

其中的符号与值的相应含义可在附录中查到。这三条规则分别为政府提供了以下的行动建议：

“如果将 DIAFINSUP 从级别 1 变为 0，那么 DOMORGVIOLENCE 的级别将从 2 以 9%的概率变为 0，或以 91%的概率保持不变；TRANSVIOLTARG 的级别将从 2 以 4%的概率变为 0，或以 7%的概率变为 3，或以 4%的概率变为 4，或以 85%的概率保持不变；TRANSVIOLOC 的级别将从 5 以 33%的概率变为级别 0，或以 7%的概率变为 2，或以 7%的概率变为 3，或以 13%的概率变为 4，或以 4%的概率保持不变。该行动建议的期望效用为 0.07。”

“如果将 DIAFINSUP 从级别 1 变为 0，同时将 ORGCULTGR 从级别 0 变为 2，那么 DOMORGVIOLENCE 的级别将从 2 以 30%的概率变为 0，或以 70%的概率保持不变；TRANSVIOLTARG 的级别将从 2 以 10%的概率变为 0，或以 20%的概率变为 3，或以 10%的概率变为 4，或以 60%的概率保持不变；TRANSVIOLOC 的级别将从 5 以 60%的概率变为级别 0，或以 40%的概率保持不变。该行动建议的期望效用为 0.12。”

“如果将 ORGCULTGR 从级别 0 变为 2，那么，DOMORGVIOLENCE 的级别将从 2 以 30%的概率变为 0，或以 70%的概率保持不变；TRANSVIOLTARG 的级别将从 2 以 11%的概率变为 0，或以 21%的概率变为 3，或以 11%的概率变为 4，或以 57%的概率保持不变；TRANSVIOLOC 的级别将从 5 以 58%的概率变为级别 0，或以 42%的概率保持不变。该行动建议的期望效用为 0.07。”

实验结果显示了算法 MABRs 的有效性。从表 3-2 中可以看到，0 位于实际效用与 MABRs 估计的期望效用的差的置信区间的正中间。同时，可以看到算法 MABRs 要比基准方法有效得多。首先，0 位于实际效用与基准方法估计的期望效用的差的置信区间之外，其次，实际效用与 MABRs 估计的期望效用的平均绝对误差大约只有与基准方法估计的期望效用的平均绝对误差的 1/2。

5. MABR-1 和 MABR-2 的比较

图 3-2 比较了 MABR-1 与 MABR-2 在某组织子数据集上当 minsup 从 8 增加到 23 时的运行时间。在另两个子数据集上的结果因为比较类似，所以在此省略。参数 minutil 对实验结果是无关紧要的。

从图 3-2 可以看出，MABR-2 效率明显较高，当 minsup 比较低时尤其如此。当 minsup 为 8 时，MABR-2 的运行时间只有 MABR-1 的 7%；当 minsup 逐渐增大时，优势逐渐变小。这是因为 minsup 越大，优势比较大的频繁行动集越少。

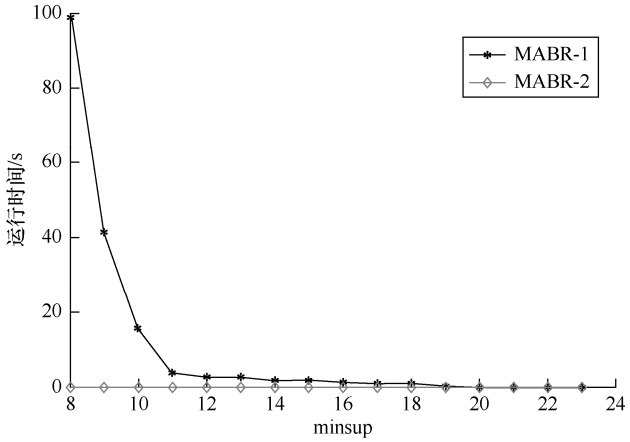


图 3-2　MABR-1 与 MABR-2 的运行时间

3.4　讨论

有人说，环境属性和行为属性的值的变化所产生的效用值对用户来讲都是难以确定的。例如，对政府来说，削减来自境外的对恐怖组织的财政支持到底要花费多大代价呢？可以通过为属性值变化指定一个效用区间而不是一个值来解决。具体来说，可以为行动 t 指定效用区间 $(\text{util}_l(t), \text{util}_h(t))$，也可以为效果 e 指定效用区间 $(\text{util}_l(e), \text{util}_h(e))$。这样，合并可操作行为规则 $r=(S,C)$ 的期望效用就定义为一个区间 $(\text{util}_l(r), \text{util}_h(r))$，其中，$\text{util}_l(r)=\sum_{t\in S}\text{util}_l(t)+\sum_{ep\in C}\text{util}_l(\text{ep.e})\times\text{ep.p}$，$\text{util}_h(r)=\sum_{t\in S}\text{util}_h(t)+\sum_{ep\in C}\text{util}_h\ (\text{ep.e})\times\text{ep.p}$。这样，当 $\text{util}_l(r)>\text{minutil}$ 或 $(\text{util}_l(r)+\text{util}_h(r))/2>\text{minutil}$ 时，可称合并可操作行为规则 r 为关于 minutil 的有趣可操作行为规则。

算法 MABR-1 和 MABR-2 假定行为信息系统是完整的。但是，在实际应用中信息系统经常不完整，例如，值的缺失并不鲜见。缺失值可能导致算法产生低质量的可操作行为规则，因此，必须设计有效的方法处理这一情况。可以采用多种缺失值处理方法或其的组合，使不完整的行为信息系统变得完整。例如，可以把行为信息系统中包含缺失值的观察删除，这是最简单的方法。其主要优点是快速并易于实现，主要缺点是会损失有价值的信息。因此，这一方法仅适合缺失值很少的情况。另外，缺失值可由行为信息系统中的其他观察的相应值填充。换句话说，如果行为信息系统中有两个观察相似，则其中一个观察的缺失值可以用另外一个观察的相应值填充。最后，还可以用行为信息系统中相应属性的平均值填充缺失值。

到现在为止，对行为信息系统的一个假设是观察的属性值要么已知，要么

未知。也就是说，观察的属性值要么是一个单值，要么是空。然而，有些时候没有足够的知识确定某观察的某属性值。例如，某观察的某属性值可能是{（进行军事打击，0.5），（不进行军事打击，0.5）}，含义是进行军事打击和不进行军事打击的概率均为 50%。为处理这一情况，需要修改可操作行为规则的支持度和置信度的定义。

以往的关于可操作规则挖掘的工作都没有经验验证提出方法的有效性。这样，挖掘出的规则的质量只能由领域专家评价。本章通过提出新的评价指标，给出了 MABRs 的经验验证，这无疑也给其他相关工作以启发。

从更广的视角来看，一些基于智能体的方法（如 POMDP）或框架（如 SOAR）也可以用效用函数和组织的过去行为来确定行动。但是，MABRs 与它们显然不同，因为其致力于发现对决策者有最佳的整体效用的行动，而不是用以达到目标状态的行动。

第 4 章

可操作行为规则挖掘技术的深入探讨

4.1 消解规则冲突

4.2 规则支持度建模

4.3 数值型行为属性建模

4.4 基于贝叶斯网络的挖掘算法

4.5 基于决策树的挖掘算法

4.6 技术展望

第 3 章提出的可操作行为规则挖掘技术是一种非常重要的组织行为模式挖掘新技术。经过验证，所提出的 MABR-1 与 MABR-2 是两种非常有效的可操作行为规则挖掘算法。但是，它们并不十分完善，甚至在一些情况下有比较大的局限性。例如，它们只能处理类别属性。对数值属性，它们采取提前离散化的处理方法，这将损失一些有意义的行为规则。再如，它们假定不同观察对规则具有相同的支持强度。在很多情况下，不同的观察具有不同的时序特征，因而对规则的支持强度也不相同。因此，这将使 MABR-1 与 MABR-2 损失一定的精确度。为使可操作行为规则挖掘技术解决上述问题，本章将致力于探讨建立精确的可操作行为规则挖掘的计算模型，设计有效、高效的规则挖掘算法。

4.1 消解规则冲突

4.1.1 规则冲突

因为历史行为数据的条件属性间通常是高度相关的，所以经常会出现这种情况：若干条候选可操作行为规则建议的行动相同，但规则形式及规则的期望效用不同。我们称这些候选可操作行为规则是冲突的。显然，我们希望保留冲突规则中具有最准确期望效用的规则而舍弃其余规则。因此，问题的关键是如何准确估计冲突规则的期望效用。

算法 MABR-1 和 MABR-2 提出了一种规则剪枝（Rule Pruning）方法，以处理冲突规则。首先，所有前件为其相应的具体规则的一部分的一般规则被剪枝。然后，包含相同行动的规则根据其支持度被合并为一条新规则。新规则的支持度实际上是被合并规则的支持度的加权平均。这种方法使用规则前件长度

作为剪枝指标。但是，这一指标仅被用于部分冲突规则。这会损害该方法的有效性。

为了克服上述缺点，本节提出了一种消解规则冲突的规则排序（Rule Ranking）方法，具体提出了一个线性组合支持度和规则前件的集成指标，以评价冲突规则的期望效用的准确性。进一步地，本节在这一集成指标中引入了一个可调的权重参数，以增加集成的灵活性。实验结果证实了该指标及方法的有效性与优越性。

本节提出的新方法受到了关联分类（Associative Classification）中的规则剪枝技术的启发。大部分关联分类算法偏爱具有更大支持度及置信度的规则。一些算法（如文献[7,8]）偏爱一般（前件较短）规则，而这会导致较低的分类准确率。相反，其他的算法（如文献[9,10]）偏爱具体（前件较长）规则，这减少了误分类概率。Thabtah 认为，规则排序指标不应仅限于支持度及置信度，并进一步提出了一种新的排序过程，这一过程在考虑支持度、置信度及前件长度之后考虑了每条规则的类分布频率。

4.1.2　冲突消解方法

1. 规则排序的候选指标

支持度与规则前件长度（规则行动集的基数）是两个评价冲突规则期望效用预测准确度的主要指标。假定其他条件相同，规则的支持度越高，则其期望效用预测的准确度越高，这是因为较高的支持度降低了规则的偶然性；规则的前件越长，则其期望效用预测的准确度越高，这是因为较长的前件为效用预测提供了更多的证据。

然而，上述两个指标经常冲突或互斥。也就是说，相对于其他规则，一条规则很可能同时具有较长的规则前件长度和较低的支持度。另外，通常并不知道这两个指标的优先级。这两个指标的优先级会随数据集的不同而不同。

2. 规则排序策略

为解决上述两个规则排序指标的冲突问题，本节使用一个可调的权重参数把它们线性组合为一个混合指标：

$$\text{score} = \alpha \cdot \sup(\text{r.S})/(|O|-1) + (1-\alpha)\cdot |r.S|/|D|, 0 \leqslant \alpha \leqslant 1$$

其中，r.S、O、D 和 α 分别表示规则 r 的前件（行动集）、行为观察集、环境属性集和可调的权重参数。

当 α=0 时，score 完全依赖规则前件长度；当 α=1 时，score 完全依赖支持度；当 $0<\alpha<1$ 时，score 依赖这两个指标的组合。α 越大，支持度在 score 中所占比重越大；α 越小，规则前件长度在 score 中所占比重越大。

若冲突规则中有几条规则有相同的最高 score 值，则可以随机保留一条规则，同时舍弃其他规则。这种策略在某些情况下效果并不好。因此，若几条规则有相同的 score 值的情况比较普遍，则需要设计另外的合理指标对这几条规则进行排序。

具体地，采用行为信息系统中的类概率分布作为对具有相同 score 值的多条规则进行排序的指标。一条规则的效果的相关属性值在行为信息系统中出现的频率越高，其排序越靠前。该指标称为 general，其计算过程如下：

输入：$I=(O,o^*,A,D,\rho)$，$r=(S,C)$：可操作行为规则

输出：r 的 general 指标值

```
38.  general←0
39.  for each  o ∈ O
40.    for each  ep ∈ C
41.      if  ρ(o,ep.e.a) = ep.e.v_t
42.        ep.sup ← ep.sup + 1
43.  for each  ep ∈ C
44.    general ← general+ | ep.p − ep.sup/ | O ‖
45.  return general
```

4.1.3　模型验证

在本节中，实验验证了提出的规则冲突消解方法的有效性。实验的目的是回答这五个问题：①新方法是否有效？②α 取何值时新方法优于老方法？③采用 score-general 联合指标是否优于采用单一 score 指标？④α 取何值时新方法优于随机选择方法？⑤当 α 值增加或减小时，新方法的效果如何变化？

1. 实验设计

从 MAROB 数据集中抽取出关于三个组织的三个行为信息系统以验证新方法。所有可能的行动和效果的效用值由领域专家指定并被规格化到区间[−1, 1]。另外，使用集合 $\{0, 0.1, 0.2, \cdots, 1\}$ 代表 α 的取值范围[0, 1]。

2. 评价指标

仍然使用平均绝对误差（MAE）（见 3.3 节）作为衡量本方法性能及不同方法优劣的评价指标。领域专家指定 MAE 的有效性阈值为 0.07。也就是说，若某方法的 MAE 值低于 0.07，则该方法有效。

3. 实验结果

表 4-1 和表 4-2 展示了新方法与其他方法在三个行为信息系统中 25 个实际发生的行动集上的实验结果。注意，其他任一实际发生的行动集均不是任一候选规则的前件。表 4-1 展示了实际效用及以往方法、随机选择法和采用 score 指标的新方法的绝对差。表 4-2 展示了实际效用以及以往方法、随机选择法和采用 score-general 联合指标的新方法的绝对差。以往方法与随机选择法的最小支持度设为 7。

表 4-1　实验结果（采用 score 指标的新方法与其他方法的比较）

		绝对差											
	以往	随机	采用 score 指标的新方法										
	方法	选择	α=0	α=0.1	α=0.2	α=0.3	α=0.4	α=0.5	α=0.6	α=0.7	α=0.8	α=0.9	α=1
1	0.0500	0.0651	0.0489	0.0400	0.0926	0.0926	0.0926	0.0926	0.0926	0.0880	0.0880	0.0880	0.0880
2	0.0471	0.0619	0.0457	0.0400	0.0926	0.0926	0.0926	0.0926	0.0926	0.0880	0.0880	0.0880	0.0880
3	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933
4	0.1244	0.1182	0.1244	0.1244	0.1244	0.1244	0.1244	0.1244	0.1244	0.1120	0.1120	0.1120	0.1120
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0500	0.0412	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0356	0.0356	0.0240	0.0240
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467
11	0.0267	0.0386	0.0267	0.0267	0.0267	0.0267	0.0267	0.0267	0.0400	0.0533	0.0533	0.0640	0.0640
12	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933
13	0.0267	0.0386	0.0267	0.0267	0.0267	0.0267	0.0267	0.0267	0.0400	0.0533	0.0533	0.0640	0.0640
14	0.0000	0.0134	0.0000	0.0000	0.0000	0.0275	0.0275	0.0275	0.0275	0.0259	0.0259	0.0289	0.0289
15	0.0000	0.0144	0.0000	0.0000	0.0000	0.0314	0.0314	0.0259	0.0259	0.0259	0.0259	0.0289	0.0289
16	0.0200	0.0234	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0300	0.0300
17	0.0000	0.0016	0.0000	0.0000	0.0000	0.0000	0.0000	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057
18	0.0200	0.0273	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0343	0.0343	0.0300	0.0300
19	0.0200	0.0198	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0171	0.0171
20	0.0933	0.0873	0.0933	0.0914	0.0914	0.0914	0.0800	0.0800	0.0800	0.0800	0.0800	0.0800	0.0800
21	0.0462	0.0455	0.0462	0.0457	0.0457	0.0457	0.0457	0.0440	0.0440	0.0440	0.0440	0.0440	0.0440
22	0.0171	0.0207	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0300	0.0300
23	0.0629	0.0809	0.0629	0.0629	0.0629	0.0629	0.1057	0.1057	0.1057	0.1057	0.0933	0.0933	0.0933
24	0.0006	0.0026	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0086	0.0086	0.0086
25	0.1371	0.1252	0.1371	0.1371	0.1371	0.1371	0.1371	0.1371	0.1057	0.1057	0.0933	0.0933	0.0933
26	0.0790	0.0773	0.0790	0.0790	0.0790	0.0790	0.0790	0.0854	0.0854	0.0854	0.0950	0.0950	0.0950
均值	0.0465	0.0496	0.0464	0.0457	0.0498	0.0520	0.0532	0.0534	0.0532	0.0534	0.0531	0.0543	0.0543

表 4-2　实验结果（采用 score-general 联合指标的新方法与其他方法的比较）

		绝对差											
	以往	随机	联合采用 score-general 联合指标的新方法										
	方法	选择	α=0	α=0.1	α=0.2	α=0.3	α=0.4	α=0.5	α=0.6	α=0.7	α=0.8	α=0.9	α=1
1	0.0500	0.0651	0.0400	0.0400	0.0926	0.0926	0.0926	0.0926	0.0926	0.0880	0.0880	0.0880	0.0880
2	0.0471	0.0619	0.0400	0.0400	0.0926	0.0926	0.0926	0.0926	0.0926	0.0880	0.0880	0.0880	0.0880
3	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933
4	0.1244	0.1182	0.1244	0.1244	0.1244	0.1244	0.1244	0.1244	0.1244	0.1120	0.1120	0.1120	0.1120
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0500	0.0412	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0356	0.0356	0.0240	0.0240
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467	0.1467
11	0.0267	0.0386	0.0267	0.0267	0.0267	0.0267	0.0267	0.0267	0.0400	0.0533	0.0533	0.0640	0.0640
12	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933	0.0933
13	0.0267	0.0386	0.0267	0.0267	0.0267	0.0267	0.0267	0.0267	0.0400	0.0533	0.0533	0.0640	0.0640
14	0.0000	0.0134	0.0000	0.0000	0.0000	0.0275	0.0275	0.0275	0.0275	0.0259	0.0259	0.0289	0.0289
15	0.0000	0.0144	0.0000	0.0000	0.0000	0.0314	0.0314	0.0259	0.0259	0.0259	0.0259	0.0289	0.0289
16	0.0200	0.0234	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0300	0.0300
17	0.0000	0.0016	0.0000	0.0000	0.0000	0.0000	0.0000	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057
18	0.0200	0.0273	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0343	0.0343	0.0300	0.0300
19	0.0200	0.0198	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0200	0.0171	0.0171
20	0.0933	0.0873	0.0933	0.0914	0.0914	0.0914	0.0800	0.0800	0.0800	0.0800	0.0800	0.0800	0.0800
21	0.0462	0.0455	0.0467	0.0457	0.0457	0.0457	0.0457	0.0440	0.0440	0.0440	0.0440	0.0440	0.0440
22	0.0171	0.0207	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0171	0.0300	0.0300
23	0.0629	0.0809	0.0629	0.0629	0.0629	0.0629	0.1057	0.1057	0.1057	0.1057	0.0933	0.0933	0.0933
24	0.0006	0.0026	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0086	0.0086	0.0086
25	0.1371	0.1252	0.1371	0.1371	0.1371	0.1371	0.1371	0.1371	0.1057	0.1057	0.0933	0.0933	0.0933
26	0.0790	0.0773	0.0790	0.0790	0.0790	0.0790	0.0790	0.0854	0.0854	0.0854	0.0950	0.0950	0.0950
均值	0.0465	0.0496	0.0458	0.0457	0.0498	0.0520	0.0532	0.0534	0.0532	0.0534	0.0531	0.0543	0.0543

可以从表 4-1 和表 4-2 看出，提出的新方法的 MAE 值明显小于有效性阈

值 0.07。这证明了提出的新方法是有效的。进一步地，当 α 设为 0 或 0.1 时，提出的新方法优于以往方法与随机选择法；当 α 设为 0.1 时，提出的新方法性能最好；当 α 设为 0 时，采用 score-general 联合指标的方法优于采用 score 指标的方法；当 α 设为其他值时，两种新方法性能相同。这是因为这些与 α 相关的具有最高 MAE 值的规则只有一条。一般地，当 α 值增加时，MAE 值单调增加。这表明规则前件长度对 score 指标的贡献明显大于对支持度的贡献，主要原因可能是较小的样本数导致规则的支持度较低，因此不同支持度之间的差异并不明显。

4.2　规则支持度建模

4.2.1　样本对规则的非一致支持强度

传统的可操作行为规则挖掘假定不同观察对规则的支持强度是相同的。而在实际领域中，不同观察具有不同的时序特征，这导致其对规则的支持强度也不相同。准确估计观察对可操作行为规则的支持强度是准确估计规则置信度的前提，也是保证规则质量的关键要素。

一般地，其时序特征距离当前时间越久的观察对规则的支持强度越小。但观察的时序特征与其对规则的支持强度之间的定量关系尚未被研究与揭示。为此，本节建立了精确的观察对可操作行为规则的非一致支持强度的数学模型。

具体地，本节提出了一个规则支持度的观察加权模型和一个相应的挖掘算法。这个观察加权模型实际上是一个线性函数，其很好地拟合了观察支持度权重与时序特征距离当前时间的距离之间的实际函数关系。基于这个线性函数，本节进一步重定义了规则的支持度及效果—概率的支持度。

4.2.2　支持度的观察加权模型

观察 o 的相关阶段与当前阶段的距离越小，则其对某可操作行为规则 r 的支持强度越大。因此，本节的任务是构建一个恰当的函数，其能很好地拟合 o 的支持度权重与其相关阶段和当前阶段的距离之间的实际函数关系。

$I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。假定 d_o 是 o 的相关阶段与当前阶段的距离，$D_{\max}=\mathrm{MAX}\{d_o \mid o\in O\}$，$D_{\min}=\mathrm{MIN}\{d_o \mid o\in O\}$，$s_o$ 是 o 的支持度权重，当 d_o=$D_{\max}$ 时，$s'=s_o$。为很好地拟合 s_o 与 d_o 之间的函数关系，需构建的函数 $\mathcal{F}$ 应满足以下三个约束条件：

① $\mathcal{F}$ 单调减少；

② $\mathcal{F}(D_{\max})=s'$，$(0<s'<1)$；

③ $\mathcal{F}(D_{\min})=1$。

线性函数具有简单易操作的特点，而且满足上述三个约束条件。因此，使用线性函数构建 $\mathcal{F}$：

$$s_o=\mathcal{F}(d_o)=\left(\frac{D_{\max}-d_o}{D_{\max}-D_{\min}}\right)\cdot(1-s')+s' \tag{4-1}$$

注意，若 $s'=1$，则 s_o=1。因此，定义 3-2 中的规则支持度模型是本节提出的观察加权模型的一个特例。

基于 $\mathcal{F}$ 的定义，可以得到以下定义：

定义 4-1 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。r=(S, C)是一个可操作行为规则。r 的加权支持度定义为

$$\mathrm{wsup}(r)=\mathrm{wsup}(S)=\sum_{o\in O}(s_o\cdot\gamma)$$

其中，若 o 支持 r，则 $\gamma=1$；否则，$\gamma=0$。

假定 minwsup 是为用户指定的一个支持度阈值，若 $\mathrm{wsup}(S)\geqslant\mathrm{minwsup}$，则称 S 为加权频繁行动集或加权频繁|S|行动集。

定义 4-2 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。$\mathrm{ep}=(e,p)$ 是一个效果−概率。若 $\rho(o,e.a)=e.v_t$，则称观察 o 支持 ep。效果−概率 ep 的加权支持度定义为

$$\mathrm{wsup}(\mathrm{ep})=\sum_{o\in O}(s_o\cdot\gamma)$$

其中，若 o 支持 ep，则 $\gamma=1$；否则，$\gamma=0$。

4.2.3　MABR-3 算法

基于 4.2.2 节提出的支持度加权模型，本节提出了一个新的可操作行为规则挖掘算法 MABR-3（见下文框中内容）。MABR-3 包含三个阶段：候选规则产生阶段、规则剪枝阶段与有趣规则产生阶段。

输入：$I=(O,o^*,A,D,\rho)$，所有可能的行动和效果的效用，minsup，minutil，s'

输出：有趣的可操作行为规则及其期望效用

// 第 1 阶段：产生候选规则

1. $F_1 \leftarrow \text{Select}(\{\text{1-action sets}\})$
2. $k \leftarrow 1$
3. **while** $F_k \neq \varnothing$
4. $F_{k+1} \leftarrow \text{Generate}(F_k)$
5. $F_{k+1} \leftarrow \text{Select}(F_{k+1})$
6. $k \leftarrow k+1$
7. $F \leftarrow \bigcup_{i=1}^{k-1} F_i$
8. **for each** $S \in F$
9. $cr \leftarrow \text{CR\_Construct}(S)$
10. $CR \leftarrow CR \cup \{cr\}$

// 第2阶段：规则剪枝

11. $R \leftarrow$ select all most specific rules from CR

// 第3阶段：产生有趣的可操作行为规则

12. **for each** $r \in R$
13. **if** $\text{util}(r) \geqslant \text{minutil}$

```
14.          IR ← IR ⋃ {(r, util(r))}
15.    return IR
16.    Function Generate(F: set of action sets)
17.    for each  {S1,S2} ⊂ F，满足 such that | S1 \ S2 |= 1
18.       if  S1 ⋃ S2  是一个行动集
19.          C ← C ⋃ {S1 ⋃ S2}
20.    for each  c ∈ C
21.       flag ← 1
22.       for each  c′ ⊂ c，满足 | c \ c′ |= 1
23.          if  c′ ∉ F
24.             flag ← 0
25.             break
26.       if  flag = 0
27.          C ← C \ {c}
28.    return C
29.    Function Select(C: set of action sets)
30.    for each  o ∈ O
31.       Co ← {c ∈ C | o supports c}
32.       for each  c ∈ Co
33.          c.wsup ← c.wsup + so
34.    for each  c ∈ C
35.       if  c.wsup < minwsup
36.          C ← C \ {c}
37.    return C
38.    Function CR_Construct(S: action set)
39.    for each  α ∈ Abe
40.       for each  v ∈ Da
```

41.　　$e \leftarrow (a, p(o^*, a), v)$

42.　　$C \leftarrow C \cup \{(e, wsup(ep) / wsup(S))\}$

43.　**return** (S,C)

4.2.4　模型验证

实验的目的是回答三个问题：①提出的观察加权方法是否有效？②新方法对于哪些 s' 的取值优于以往的方法？③当 s' 由 1 单调减少至 0 时，新方法的性能将如何变化？

1. 实验设计

从 MAROB 数据集中抽取出关于三个组织的三个行为信息系统以验证新方法。所有可能的行动和效果的效用值由领域专家指定并被规格化到区间[−1, 1]。另外，使用集合 $\{0.1, 0.2, \cdots, 1\}$ 代表 s' 的取值范围 $(0,1]$。

2. 评价指标

本节仍然使用平均绝对误差（MAE）（见 3.3 节）作为衡量本方法性能及不同方法优劣的评价指标。

3. 实验结果

图 4-1 展示了当 minwsup 设为 3 时，对不同 s' 值新方法在三个 MAROB 子数据集上的平均绝对误差。当 s' 由 0.1 增加到 1 时，MAE 值单调增加。这证明了新方法的有效性和优越性。当 $s' = 0.1$ 时，新方法性能最好；当 $s' = 1$ 时，新方法性能最差。注意：当 $s' = 1$ 时，新方法等价于已有方法。

对于本试验所采用的三个数据集，MAE 是单调增加的。但对于其他领域未必如此。因此，对不同领域需要实验选择不同的使新方法性能最优的 s' 值。

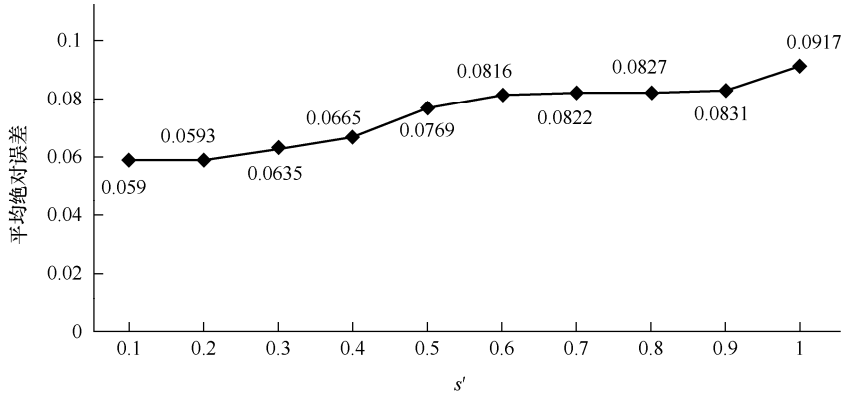

0.1
0.08
0.06
0.04
0.02
0
平均绝对误差
0.059
0.0593
0.0635
0.0665
0.0769
0.0816
0.0822
0.0827
0.0831
0.0917
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
s'

图 4-1　平均绝对误差

4.3　数值型行为属性建模

4.3.1　问题的提出

可操作行为规则的传统定义假定所有行为属性都是类别属性。如果存在数值属性，则提前对其进行离散化处理。这一假定会损害 MABR-1 与 MABR-2 算法的有效性。为解决这一问题，本节提出了直接基于数值行为属性的可操作行为规则挖掘的新定义，以及一个相应的新的挖掘算法（MABR-4）。实验结果证实了提出的新定义和新方法的有效性。

4.3.2　问题定义

本节重定义了可操作行为规则挖掘问题。

定义 4-3　关于某组织的行为信息系统（Behavioral Information System）定义为一个 5 元组 $I=(O,o^*,A,D,\rho)$，其中 O 是对实体（组织）的观察的有限非空集，$o^*\in O$ 是下一个观察的投射，A 是属性的有限非空集，$D=\bigcup_{a\in A}D_a$（D_a 是属性 a 的值域），$\rho:O\times A\to D$ 是一个将每个观察和属性值的集合关联起来的函数。A 可进一步分为两个子集，即 $A=A_{en}\bigcup A_{be}$，其中，A_{be} 是描述组织行为的行为属性的集合，A_{en} 是描述组织所处环境并对行为属性有影响的环境属性的集合。

假定所有行为属性都是数值属性，而所有环境属性都是类别属性。如果存

在数值环境属性，则提前对其进行离散化处理。

定义 4-4 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。行动（Action）定义为一个三元组 $t=(a,v_f,v_t)$，其中，$\alpha\in A_{en}$，$v_f=\rho(o^*,a)$，$v_t\in D_a$。如果 $v_f\neq v_t$，则称 t 为标准行动（Standard Action）；如果 $v_f=v_t$，则称 t 为非标准行动（non-Standard Action）。如果 a 的值从 v_f 变到 v_t，则称行动 $t=(a,v_f,v_t)$ 满足（Holds）。行动集（Action Set）S 又称为 $|S|$−行动集，定义为行动的有限非空集，其中，对任何 $t_1,t_2\in S$，有 $t_1.a\neq t_2.a$。如果每个 $t\in S$ 都满足，则称行动集 S 满足。如果对每个 $t\in S$ 有 $\rho(o,t.a)=t.v_t$，则称观察 o 支持 S。S 的支持度定义为

$$\sup(S)=\left|\{o\in O\mid o \text{ supports } S\}\right|$$

如果 $\sup(S)\geqslant$ minsup，则称 S 为关于一个用户指定的阈值——最小支持度（Minsup）的频繁行动集（Frequent Action Set）或者频繁 $|S|$−行动集。

定义 4-5 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。效果定义为一个三元组 $e=(a,v_f,v_t)$，其中，$a\in A_{be}$，$v_f=\rho(o^*,a)$，$v_t\in D_a$。如果 a 的值从 v_f 变为 v_t，则称效果 $e=(a,v_f,v_t)$ 发生。

定义 4-6 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。候选可操作行为规则（Candidate Actionable Behavioral Rule）定义为 $r=(S,C)$，其中，S 是一个行动集，C 是效果的一个有限非空集，$|C|=|A_{be}|$。可操作行为规则 $r=(S,C)$ 表示若行动集 S 满足，则效果集 C 将发生。

改变环境属性的值和行为属性的值都会为用户带来收益（正效用）或损失（负效用）。换句话说，行动会有或正或负的效用。很明显，如果环境或行为属性值没有变化，则相应的效用为 0。

定义 4-7 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。可操作行为规则 $r=(S,C)$ 的期望效用（Expected Utility）定义为

$$u(r)=\sum_{t\in S}u(t)+\sum_{e\in C}(\lambda_{e.a}\cdot \mathrm{uu}(e.a)\cdot(e.v_t-e.v_f)/\mathrm{ul}(e.a))$$

其中，$u(t)$、$\mathrm{ul}(e.a)$和$\mathrm{uu}(e.a)$分别表示行动 t 的效用、$D_{e.a}$ 的单位长度和效果 e' 的效用，$e'.a=e.a$ 且$|e'.v_t-e'.v_f|=ul$。若用户偏爱较大的 $e.a$ 值，则 $\lambda_{e.a}=1$，否则 $\lambda_e=-1$。如果 $u\ (r)\geqslant$minutil，则称可操作行为规则 r 为关于用户指定最小效用（Minutil）阈值的有趣可操作行为规则（Interesting Actionable Behavioral Rule）。

4.3.3　MABR-4 算法

基于 4.3.2 节中的新定义，本节提出了一个新的可操作行为规则挖掘算法 MABR-4（见下文框中内容）。MABR-4 包含三个阶段：候选规则产生阶段、规则剪枝阶段与有趣规则产生阶段。

在候选规则产生阶段，首先产生所有的频繁行动集，然后以频繁行动集为前件产生候选规则（通过调用第 3 行中的 CR_Construct 函数）。候选规则阶段的输出是候选可操作行为规则集合。在规则剪枝阶段，采用 4.1 节提出的规则剪枝方法消解候选可操作行为规则中的规则冲突。有趣规则产生阶段从剪枝后的候选规则中产生有趣的可操作行为规则。

输入：$I=(O,o^*,A,D,\rho)$，所有可能的行动和效果的效用，$\{uu(a)\,|\,a\in A_{bc}\}$，minsup，minutil

输出：有趣的可操作行为规则及其期望效用

// 第 1 阶段：产生候选规则

1.　F ← 所有的频繁行动集

2.　**for each**　$S\in F$

```
3.        cr ← CR _ Construct(S)
4.        CR ← CR ∪ {cr}

// 第 2 阶段：规则剪枝
5.    for each 关系～在 CR 上的最大等价类 LE
6.       r←随机选择一条其行动集的基数最大的规则
7.        R ← R ∪ {r}

// 第 3 阶段：产生有趣的可操作行为规则
8.    for each   r ∈ R
9.        if   u(r) ≥ minutil
10.           IR ← IR ∪ {(r, util(r))}
```

4.3.4 模型验证

1. 数据集

本节使用 Vodka 数据集[①]验证提出的新定义及 MABR-4 算法的有效性。该数据集记录了美国 31 个伏特加品牌 15 年的年销售量、价格及广告支出等数据。我们从中抽取了与 31 个品牌相关的 31 个行为信息系统。环境属性包括年杂志广告支出、年报刊广告支出、年户外广告支出和年广播广告支出。这些数值环境属性均被离散化为取值范围为{1, 2, 3}的类别属性。市场份额是唯一的行为属性。另外，可以从数据集中抽取所有可能行动的效用和效果的单位效用。

表 4-3 描述了品牌 Smirnoff 相关的行为信息系统的 ρ 函数。

① http://faculty.darden.virginia.edu/

表 4-3　品牌 Smirnoff 相关的行为信息系统的 ρ 函数

	年	广告支出				市场份额
		杂志	报刊	户外	广播	
O	1995	3	1	2	1	0.246
	1996	3	3	3	1	0.253
	1997	3	1	3	1	0.237
	1998	3	3	2	1	0.270
	1999	2	2	3	1	0.244
	2000	2	3	1	3	0.218
	2001	2	2	3	2	0.227
	2002	1	1	3	3	0.233
	2003	2	1	1	3	0.233
	2004	1	1	1	2	0.217
	2005	1	1	1	2	0.224
	2006	1	1	1	2	0.218
	2007	1	1	1	2	0.217
	2008	1	1	1	3	0.230
o^*	2009	1	1	1	2	0.235

2. 基准方法

可操作行为规则 $r=(S,C)$ 表示若行动集 S 满足，则效果集 C 将发生。对一个行动集，不同的方法会产生不同的效果集。换句话说，对一个行为属性 a，不同的方法会得到不同的效果。

假定一个行动集满足，可以计算 $\{\rho(o,e.a)\mid o\in O\}$ 的均值作为 $e.v_t$ 来估计 e。基于此，基准方法以这样的策略来构建可操作行为规则：对任一行动集 S，规则 $r=(S,C)$ 被构建，其中，对任一 $a\in A_{be}$，存在一个效果 $((a,\rho(o^*,a),v),\sum_{o\in O}\rho(o,e.a)/|O|)\in C$。

3. 评价指标

本节仍然使用平均绝对误差（MAE）（见 3.3 节）作为衡量本方法性能及不

同方法优劣的评价指标。领域专家指定 MAE 的有效性阈值为 0.01。也就是说，若某方法的 MAE 值低于 0.01，则该方法有效。

4. 实验结果

表 4-4 展示了 MABR-4 算法与基准方法的实验结果对比。具体来说，其展示了当最小支持度依次设为 1～9 时基准方法和 MABR-4 算法的绝对差的均值和标准差。

表 4-4　MABR-4 算法与基准方法的实验结果对比

minsup	绝对误差			
	均值		方差	
	MABR-4	基准方法	MABR-4	基准方法
1	0.0097	0.0253	0.0121	0.0400
2	0.0098	0.0249	0.0125	0.0450
3	0.0097	0.0262	0.0132	0.0508
4	0.0083	0.0297	0.0100	0.0566
5	0.0074	0.0115	0.0097	0.0137
6	0.0068	0.0129	0.0098	0.0156
7	0.0073	0.0128	0.0105	0.0174
8	0.0069	0.0130	0.0110	0.0183
9	0.0081	0.0157	0.0121	0.0197

实验结果证实了我们提出的方法的有效性。从表 4-4 可以看出，无论最小支持度取何值，MABR-4 算法的 MAE 值都小于有效性阈值 0.01。当最小支持度取 6 时，MABR-4 算法效果最好。当最小支持度自 6 增加时，MABR-4 算法的性能逐渐变差，原因是过低的最小支持度导致了规则的偶然性。最小支持度自 6 减小时，MABR-4 算法的性能也逐渐变差，原因是过高的最小支持度导致过小的规则行动集的基数，进一步导致支持规则的证据不足。

从表 4-4 中还可以看出 MABR-4 算法的 MAE 值显著小于基准方法的 MAE 值。特别地，当最小支持度取 6 时，基准方法的 MAE 值几乎是 MABR-4 算法

的 MAE 值的 2 倍。这表明 MABR-4 算法的性能显著优于基准方法。

注意：MAE 值随最小支持度的不同而不同，原因是对于某个最小支持度，MABR-4 算法可以产生一条以某一实际行动集为标准行动集的规则；而对于一个更高的最小支持度，其有可能无法产生以该实际行动集为标准行动集的规则。对前一种情况，两种方法的 MAE 值都包含在实验结果中；对后一种情况，其 MAE 值并未含入实验结果。

4.4 基于贝叶斯网络的挖掘算法

4.4.1 问题的提出

以往的可操作行为规则挖掘算法为计算行动集的支持度需要多次重复扫描行为数据集，这一过程非常耗时。若一个行为信息系统的属性较多且（或）最小支持度较低，那么这些算法会耗费非常多的运行时间，甚至会影响可行性。

为解决上述问题，本节提出了一个基于贝叶斯网络（Bayesian Network）的可操作行为规则挖掘方法及相应算法。实验结果证实了提出的新方法的有效性与优越性。

4.4.2 贝叶斯网络

贝叶斯网络又称信念网络（Belief Network）或有向无环图模型（Directed Acyclic Graphical Model），是目前不确定知识表达和推理领域最有效的理论模型之一。贝叶斯网络反映了部分随机变量的状态，并用概率描述了这些变量的联系，适用于表达和分析不确定性和概率性的事件，应用于有条件地依赖多种控制因素的决策，可以从不完全、不精确或不确定的知识或信息中做出推理。例如，一个贝叶斯网络可以反映疾病和症状之间的概率关系，给定症状，则可被用来计算各种疾病出现的概率。

一个贝叶斯网络是一个有向无环图（Directed Acyclic Graph，DAG），由代表变量的节点集，以及连接这些节点的有向边构成。节点代表随机变量，节点间的有向边由父节点指向子节点，代表节点间的依赖关系。这些依赖关系通常是因果关系。关系强度用条件概率表达，没有父节点的节点用先验概率进行信息表达。节点变量可以是任何问题的抽象，如测试值、症状等。

除了作为贝叶斯网络模型的定性部分的有向无环图结构，还需要明确模型的定量参数。这些参数以一种与马尔科夫特性（Markovian Property）相一致的方式描述。每个节点中的条件概率分布（CPD）依赖其父节点。对离散随机变量，一般用一张表来表示相应节点的每个可能值对其父节点取值的任一可能组合的局部条件概率。变量集合的联合概率分布可被这些局部条件概率表（CPTs）唯一决定。若已知其他变量的值，则贝叶斯网络节点的任一取值的概率都是可计算的。另外，因为条件关系被有向边清楚地定义，变量之间的独立性很容易被识别，因此，不是所有的贝叶斯系统中的联合概率都需要为作决策而被计算。

基于上面的介绍，可以给出贝叶斯网络的形式化定义。一个贝叶斯网络 B 是一个表示关于一组随机变量 V 的联合概率分布的带注释的无环图。其可定义为一个二元组 $B=\langle G,\Theta\rangle$，其中，节点 $X_1,X_2,\cdots,X_n$ 表示随机变量，边表示节点之间的依赖。图 G 假设每个变量 X_i 独立于它的非子节点。Θ表示一组网络参数，包括参数 $\theta_{x_i|\pi_i}=P_B(x_i\mid\pi_i)$（$x_i$ 是变量 X_i 的一个可能取值）与一组 X_i 的父节点。相应地，B 在 V 上定义在了一个联合概率分布：

$$P_B(X_1,X_2,\cdots,X_n)=\prod_{i=1}^{n}P_B(X_i\mid\pi_i)=\theta_{X_i|\pi_i}$$

一个最简单的贝叶斯网络可由领域专家指定，随后可被用于推断。在很多应用中领域，专家很难定义网络。在这种情况下，网络结构和局部分布参数只能从数据中学习得到。为简化问题，本节中采用的贝叶斯网络由领域专家直接指定。

给定一个指定局部条件概率表的贝叶斯网络，可以通过边缘化（marginalization）评价所有可能的推断查询。有两种推断支持比较常见。第一

种通过其祖先连接到 X_i 的证据节点的对节点 X_i 的预测支持（也被称为自上而下的推理）。第二种通过其子孙连接到 X_i 的证据节点的对节点 X_i 的诊断支持（也被称为自下而上的推理）。本节采用自上而下的推理。

4.4.3 问题定义

本节关于可操作行为系统的定义大部分沿用 4.3.2 节中的相关定义，包括定义 4-1～定义 4-5。另外，本节定义了一个新的概念——影响网络。

定义 4-8 $I=(O,o^*,A,D,\rho)$ 是一个行为信息系统。关于 I 的一个影响网络（Influence Network）定义为一个贝叶斯网络 $\mathrm{IN}=\langle G,\Theta\rangle$，其中，节点表示环境和行为属性，边表示行为属性对环境属性的直接依赖。参数 Θ 是一个 CPT，列出了行为属性对父节点（环境属性）任一可能取值组合的任一可能取值的局部概率。

4.4.4 MABR-5 算法

基于 4.3 节中的新定义，本节提出了一个新的可操作行为规则挖掘算法 MABR-5（见下文框内）。MABR-5 包含两个阶段：候选规则产生阶段与有趣规则产生阶段。

输入：$\mathrm{I}=(\mathrm{O},\mathrm{o}^*,\mathrm{A},\mathrm{D},\rho)$，IN：关于 I 的影响网络，所有可能的行动和效果的效用，minsup，minutil

输出：有趣的可操作行为规则及其期望效用

// 第 1 阶段：产生候选规则

1. **for each** $\mathrm{oe}\in\prod_{\mathrm{ae}\in\mathrm{A_{en}}}\mathrm{D_{ae}}$
2. **for each** $\mathrm{ae}\in\mathrm{A_{en}}$

```
3.          if  ρ(o*,ae) ≠ ρ(oe,ae)
4.              S ← S ∪ {(ae,ρ(o*,ae),ρ(oe,ae))}
5.      for each  ab ∈ A_be
6.          for each  v ∈ D_ab
7.              e ← (ab,ρ(o*,ab),v)
8.              p ← ab 取 v 值的概率
9.              C ← C ∪ {(e,p)}
10.   CR ← CR ∪ {(S,C)}

// 第 2 阶段：产生有趣的可操作行为规则
11.   for each  r ∈ CR
12.     if  util(r) ≥ minutil
13.         IR ← IR ∪ {(r,util(r))}
14.   return IR
```

4.4.5　模型验证

1. 实验设计

本节从 MAROB 数据集中抽取关于三个组织的三个行为信息系统以验证新方法。所有可能的行动和效果的效用值由领域专家指定并被规格化到区间[−1, 1]。

2. 评价指标

本节仍然使用平均绝对误差（MAE）（见 3.3 节）作为衡量本方法性能及不同方法优劣的评价指标。领域专家指定 MAE 的有效性阈值为 0.07。也就是说，若某方法的 MAE 值低于 0.07，则该方法有效。

3. 实验结果

MABR-5 算法的 MAE 值为 0.59，明显小于有效性阈值 0.07。这说明了新方法的有效性。

另外，图 4-2 比较了算法 MABR-5 与 MABR-2 的运行时间。相比 MABR-1 等其他的挖掘算法，MABR-2 有良好的可伸缩性和效率，避免了潜在的可操作行为规则的产生—检测步骤，并使用 FA-tree 数据结构，显著地减少计算代价。

从图 4-1 中可以看出，MABR-5 的运行时间显著小于 MABR-2 的运行时间，这充分说明新方法在时间复杂度方面明显优于以往方法。

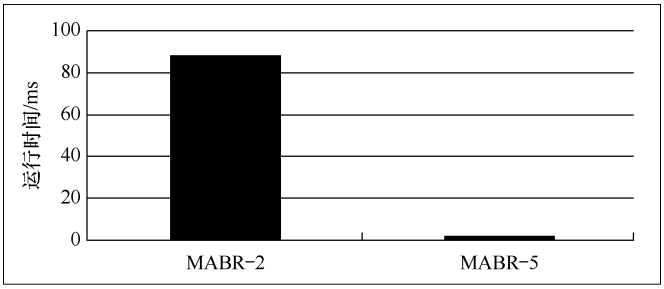


图 4-2　MABR-5 与 MABR-2 的运行时间

4.5　基于决策树的挖掘算法

4.5.1　问题的提出

当数据集海量且高维时，MABR-2 会面临大量的内存消耗、高 I/O 代价、高时空复杂度等严峻挑战。

为解决上述问题，本节提出了一个基于决策树（Decision Tree）的可操作行为规则挖掘算法。首先，以每个行为属性作为决策属性，构建决策树分类器。然后，将决策树转化为多条分类规则。最后，基于这些决策规则构建可操作行为规则。这一新方法通过避免发现频繁行动集，极大地减少了时间复杂度。实验结果表明了新方法的有效性与优越性。

4.5.2　MABR-6 算法

基于 4.3.2 节中的相关定义，本节提出了一个新的可操作行为规则挖掘算法 MABR-6（见下文框内）。MABR-6 包含两个阶段：候选规则产生阶段与有趣规则产生阶段。

输入：$I=(O,o^*,A,D,\rho)$，所有可能的行动和效果的效用，minutil

输出：有趣的可操作行为规则及其期望效用

// 第 1 阶段：产生规则

1.　**for each**　$a \in A_{be}$

```
2.      root←使用 C4.5 算法分别以 a 、A_en 为决策属性与条件属性从 I 中产生一棵决策树
3.      Generate(root)
4.  for each  S ∈ F
5.      r ← Construct(S)
6.      R ← R ⋃ {r}

// 第 2 阶段：产生有趣的可操作行为规则
7.  for each  r ∈ R
8.      if  util(r) ≥ minutil
9.          IR ← IR ⋃ {(r, util(r))}
10. return IR
11. Function Generate(Node:  决策树节点, S:  行动集)
12. if  Node.Parent ≠ null
13.     a ← Node.Parent.Attribute
14.     S ← (a, ρ(o*, a), Node.Value)
15. if Node has no child
16.     F←S
17.   S ← S \ {a}
18.     return
19. else
20. for each child node child of Node
21.     Generate(child, S)
22.  S ← S \ {a}
23. Function Construct (S: action set)
24. for each  a ∈ A_bc
25. for each  v ∈ D_a
26.     e ← (a, ρ(o*, a), v)
27.     C ← C ⋃ {(e,| {o ∈ O | o supports(S,e)} | / sup(S))}
28. return (S,C)
```

在候选规则产生阶段，首先，使用 C4.5 算法从行为信息系统 *I* 中学习得到若干棵决策树，每个行为属性作为决策属性对应其中一棵树。然后，迭代遍历每棵树得到行动集。最后，基于这些行动集构建相应的可操作行为规则。

第 2 阶段从第 1 阶段产生的候选规则中构建有趣的可操作行为规则。

4.5.3　模型验证

本节采用了一个包含 4629 个观察、216 个环境属性、1 个行为属性的模拟数据集，来验证新方法的有效性。最小支持度设为 10～100。图 4-3 中比较了算法 MABR-6 与 MABR-2 的运行时间。我们可以看到，MABR-6 的运行时间显著小于 MABR-2。这充分说明新方法在时间复杂度方面明显优于以往方法。

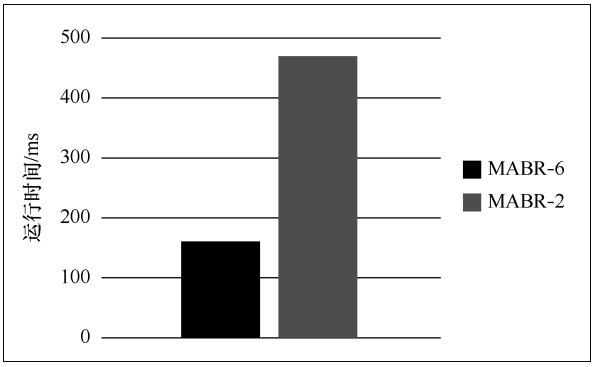


图 4-3　MABR-6 与 MABR-2 的运行时间

4.6 技术展望

4.6.1 发展方向

1. 遗传挖掘算法

遗传挖掘算法研究的一个引人注目的新动向是基于遗传算法的机器学习，它把遗传算法从历来离散的搜索空间的优化搜索算法扩展到具有独特的规则生成功能的崭新的机器学习算法。这一新的学习机制对于解决人工智能中知识获取和知识优化精炼的瓶颈难题带来了希望。因此，本节将研究把遗传算法应用于可操作行为规则挖掘。

2. 分布式挖掘算法

相比串行算法，相应的分布式挖掘算法具有并行处理的高效性特点。而行为信息系统可分为多个子集，这使得为 MABR-1 与 MABR-2 等串行算法设计相应的分布式算法成为可能。另外，由于遗传挖掘算法具有天然的并行性，因此，引入分布式处理可提高求解速度，而且种群规模的扩大和各子种群的隔离，使得种群的多样性得以丰富和保持，减少了未成熟收敛的可能性，提高了求解质量。

3. 次优挖掘算法

相比最优挖掘算法，相应的次优挖掘算法可以获得较低的时间复杂度。决策树或决策表等基于规则的分类技术使用贪心技术获得次优的分类规则。本节将研究利用基于规则的分类技术生成的分类规则产生候选可操作行为规则的高效挖掘算法。

4.6.2　发展方案

1. 遗传挖掘算法

拟采用二进制串进行编码。具体来说，假定行为信息系统中某条件属性 e 的值域为 D_e，若 $2^{i-1}<|D_e|-1\leqslant 2^i$（$i$ 为非负整数），则关于 e 的染色体段可用 i 位二进制串编码。染色体的编码由关于各条件属性的染色体段链接而成。拟将可操作行为规则的期望效用函数作为适应度函数。拟对交叉概率 P_c 和变异概率 P_m 采用自适应策略，即 P_c 和 P_m 能够随适应度自动改变。种群规模 M 拟取 20～100 的整数。终止进化代数 T 拟取 100～500 的整数。

2. 分布式挖掘算法

这里基于高性能计算集群设计分布式挖掘算法。拟采用 8 个 SMP 超节点组成集群系统，每个 SMP 超节点具有 2 个超线程技术（HT）的 INTEL Xeon2.8GHz CPU，这两个 CPU 共享 4GB 内存和 320GB 硬盘。8 个 SMP 超节点之间通过千兆以太网连接。另外，本节使用 MPI（Message Passing Interface，消息传递接口）实现各个计算节点的通信。

（1）分布式 MABR-1 算法

拟采用以下策略：①将行为信息系统 I 分为不同的部分 $I_i(i=1,\cdots,p)$ 并分配到每个处理器 p_i；②使用在第 k–1 阶段生成的全部势为 k–1 的频繁行动集生成全部势为 k 的候选行动集；③p_i 遍历 I_i 计算势为 k 的候选行动集的本地支持度；④p_i 相互交换势为 k 的候选行动集的本地支持度；⑤p_i 从势为 k 的全局候选行动集构建势为 k 的频繁行动集。

（2）分布式 MABR-2 算法

拟采用以下策略：①将行为信息系统 I 分为不同的部分 $I_i(i=1,\cdots,p)$ 并分配到每个处理器 p_i；②p_i 从 I_i 构建势为 1 的本地行动集列表；③聚合 p_i 上的本

地列表构建势为 1 的全局频繁行动集列表（GLT）；④p_i根据 GLT 产生一棵本地频繁行动集树；⑤将 GLT 的不相交部分均匀分配给每个 p_i；⑥每个 p_i 部分交换由顺序算法挖掘出频繁行动集树，以发现所有的频繁行动集。

（3）分布式遗传挖掘算法

拟采用粗糙模型，它将群体依照处理器的个数分成若干个子群体，各个子群体在各自的处理器上并发独自运行顺序算法。每经过一定的进化代，各个子群体间将交换若干个个体。迁移拓扑采用单向环拓扑结构。迁移周期采用异步迁移方式。迁移率为 10%～20%。迁移策略采用最优个体迁出/最差个体被替代的方法。

3. 次优挖掘算法族

拟采用以下策略：①利用决策树或决策表分类技术为所有行为属性的所有可能取值构建分类规则；②对任一条分类规则，若存在$|A_{be}|-1$条分类规则的前件与该分类规则相同，则使用这$|A_{be}|$条分类规则构建一条候选可操作行为规则。

第 5 章

大数据背景下的组织行为模式挖掘

5.1 大数据时代

5.2 面临的挑战

5.3 应对策略

5.4 总体目标与关键问题

5.5 实现方案

5.1 大数据时代

大数据是指以多元形式，自许多来源搜集而来的庞大数据集，往往具有实时性，并且无法用传统数据库工具对其内容进行抓取、管理和处理。大数据有“4V”特征：

（1）大容量（Volume）：数据规模一般在 10TB 左右，但在实际应用中，很多企业用户把多个数据集放在一起，已经形成了 PB 级的数据量。

（2）多样性（Variety）：数据来自多种数据源，数据种类和格式日渐丰富，已冲破了以前所限定的结构化数据范畴，囊括了半结构化和非结构化数据。

（3）快速度（Velocity）：在数据量非常庞大的情况下，也能够做到数据的实时处理。

（4）真实性（Veracity）：随着社交数据、企业内容、交易与应用数据等新数据源的兴起，传统数据源的局限被打破，企业愈发需要确保其真实性及安全性。

随着计算机存储能力的提升和复杂算法的发展，近年来的数据量呈指数型增长，这些趋势使科学技术的发展日新月异，商业模式也发生了颠覆性的变化。《分析的时代：在大数据的世界竞争》是 2016 年 12 月麦肯锡全球研究院（MGI）发表的一份报告。2011 年 MGI 就指出大数据分析在基于定位的服务、美国零售业、制造业和欧盟公共部门及美国健康医疗领域有很大的增长潜力。数据正在被商业化，来自网络、智能手机、传感器、相机、支付系统和其他途径的数据形成了一项资产，产生了巨大的商业价值。苹果、亚马逊、Facebook、谷歌、

通用、微软和阿里巴巴利用大数据分析及自己的优势改变了竞争的基础，建立了全新的商业模式。稀缺数据的所有者利用数字化网络平台在一些市场近乎实现垄断，只需用独特方式将数据整合分析，提供有价值的数据分析，几乎可以“赢家通吃”。2011 年全球的数据储量就达到了 1.8ZB，与 2011 年相比，2018 年数据储量增长了近 8 倍，未来 10 年，预计全球数据储量还将增长 10 倍，大数据成为提升产业竞争力和创新商业模式的新途径。大数据在企业中得到了充分的应用并实现了巨大的商业价值。例如，梅西百货的 SAS 系统可以根据 7300 种货品的需求和库存实现实时定价。零售业寡头沃尔玛通过最新的搜索引擎 Polaris，利用语义数据技术使得在线购物的完成率提升了 10%～15%。

5.2 面临的挑战

可操作行为规则挖掘技术在商务智能领域有广泛的应用前景。例如，因为在现代酒类产品市场中，广告投入对市场份额（消费者购买行为）的影响是决定性的，可操作行为规则挖掘可为酒类厂商提供如下行动建议：

若将户外广告投入份额由 0.15 提高至 0.2，在线广告投入份额由 0.07 提高至 0.16，电视广告投入份额由 0.12 降低至 0.08，无线电广告投入份额由 0.12 降低至 0.08，其品牌的市场份额将由 0.1 提高至 0.12，且将获得 0.8 亿元的净收益。

商业大数据时代不仅为传统的可操作行为规则挖掘技术带来了空前机遇，同时也带来了严峻挑战：

① 企业外部数据爆炸式增长，其在指导企业运营中占有的权重越来越高。而传统的可操作行为规则挖掘仅考虑企业内部数据，缺乏利用外部数据价值的有效手段。

② 在企业内外部数据中，分类数据所占的比例越来越低，数值型数据所占的比例则越来越高。而传统的可操作行为规则挖掘算法对数值型数据所采用的离散化处理方式会损失很多有意义的规则。

③ 全球所拥有的商业数据爆发式增长，总量已远远超过历史上的任何时期。传统的可操作行为规则挖掘算法在很多应用中已经难以在可接受的时间内完成计算。

因此，解决以上挑战，使传统可操作行为规则挖掘技术适应商业大数据环境，将是非常有意义的工作。

5.3　应对策略

为有效应对商业大数据环境带来的挑战，应采取的策略包含：可操作行为规则挖掘中的企业内外部数据融合；面向商业大数据的可操作行为规则挖掘建模；面向商业大数据的可操作行为规则挖掘算法设计；实验验证与应用示范。策略内容的层次关系如图 5-1 所示。

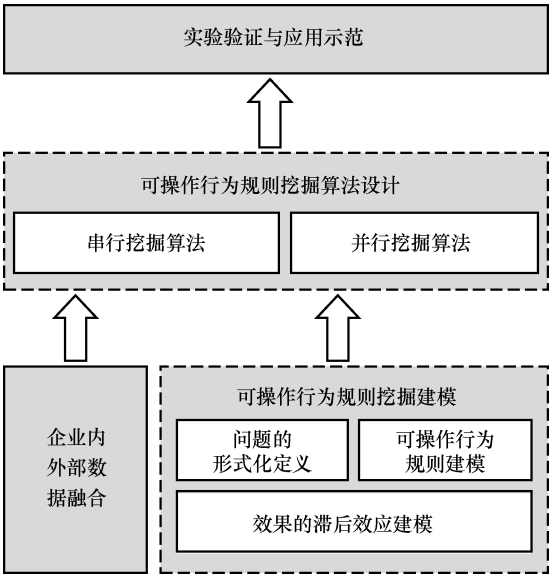


图 5-1　策略内容的层次关系

1. 可操作行为规则挖掘中的企业内外部数据融合

随着互联网、社交媒体等新技术、新应用的急速发展，企业外部数据爆发式增长，其在指导企业运营中占有的权重越来越高。企业不仅要关心内部的信

息整合，如 CRM、ERP，而且必须关心外部的相关评论、口碑、商誉、舆情、留言等。

传统的可操作行为规则挖掘仅考虑企业内部数据，要有效利用外部数据的价值还存在很多困难，如确保外部数据的可靠性，对外部非结构化和半结构化原始数据的分析与处理，互联网虚假信息的识别等。因此，有必要为可操作行为规则挖掘发展新技术、新手段，使其能够有效融合企业内外部数据，显著提高所挖掘出的知识对企业的管理决策的价值。

2. 面向商业大数据的可操作行为规则挖掘建模

在传统可操作行为规则挖掘的相关定义中，环境属性和行为属性都是分类属性。如果存在数值型属性，则事先进行离散化处理。但是离散化使得规则挖掘算法只能获得局部最优解，从而损失很多有价值的规则。随着大数据时代的到来，相对分类数据，数值型数据在商业中所占的比重越来越高。因此，可操作行为规则挖掘有必要发展新的技术与模型，直接处理数值型数据。

（1）问题的形式化定义

传统的可操作行为规则挖掘被定义为一个支持度—期望效用框架下的搜索问题。而基于数值型数据的可操作行为规则挖掘本质上是一个以期望效用为目标函数的优化问题。这就有必要为可操作行为规则挖掘建立一套全新的形式化概念体系。

（2）可操作行为规则建模

可操作行为规则挖掘的本质是研究行动和效果之间的相关、依赖关系。传统的可操作行为规则挖掘基于分类数据，其规则建模的技术基础是频繁模式发现。回归分析（Regression Analysis）是确定数值型变量间相互依赖的定量关系的一种应用最广泛的统计分析方法，十分适合作为商业数值型数据环境下可操作行为规则建模的技术基础。因此，未来将分别建立基于线性回归（Linear Regression）、支持矢量回归（Support Vector Regression）与高斯过程回归

（Gaussian Processes Regression）等回归技术的规则模型。

（3）效果的滞后效应（Time-Lag Effect）建模

效果一般存在滞后效应，也就是说，当期效果不仅受当期行动的影响，还可能受往期行动和效果的影响。因此，有必要对其进行精确建模，以加强相应规则模型的有效性。未来将分别建立基于分布滞后模型（Distributed Lag Model）、自回归模型（Autoregressive Model）与自回归分布滞后模型（Autoregressive Distributed Lag Model）的规则模型。

3. 面向商业大数据的可操作行为规则挖掘算法设计

（1）串行挖掘算法设计

未来将基于策略 2 所构建的各种规则模型设计多种有效的串行可操作行为规则挖掘算法。

（2）并行挖掘算法设计

市场与技术的发展瞬息万变，每个企业高级管理人员都必须提高自己对外界变化的快速反应能力。只有善于作出快速、正确的决策，才能将企业引向所预见的方向发展。随着商业数据量的爆发式增长，串行数据挖掘算法越来越不能满足用户对快速决策的要求，并行化成为其提升时效的最佳选择。因此，未来将为可操作行为规则挖掘的各种串行算法设计相应的多种并行方案。

4. 实验验证与应用示范

未来将建立针对客户购买行为和客户流失行为的两个应用示范，分别满足用户获取行动建议以促进产品/商品销售（增加客户购买行为），并且提高客户忠诚度（减少客户流失行为）。

5.4 总体目标与关键问题

1. 总体目标

总体目标为可操作行为规则挖掘建立新的技术内涵与方法体系，使其很好地适应大数据时代，成为大数据背景下的一种成熟的行为分析新技术。具体来说：

① 使可操作行为规则挖掘能够深度有机融合企业内外部数据，显著提升所发现知识对企业管理与决策的价值。

② 建立面向商业大数据的可操作行为规则挖掘的概念与模型体系，使其能有效处理数值型商业数据。

③ 设计多种有效、高效的面向商业大数据的可操作行为规则挖掘算法。

④ 为面向商业大数据的可操作行为规则挖掘建立应用示范。

2. 关键问题

（1）内外部数据融合中，外部数据的可靠性问题

由于行为数据集中的样本数据可能取自一个很小的时间间隔，这就导致某些时间间隔对应的外部数据因为来源样本较少甚至缺失而带来的数据可靠性问题。拟用修正的热卡填充（Adjusted Hot Deck Imputation）等方法解决该问题。

（2）效果的滞后变量模型的多重共线性问题

行动的滞后值、当期值之间，效果的滞后值、当期值之间都可能存在相关性，所以滞后变量模型一般存在多重共线性问题。拟用阿尔蒙多项式（Almon

Polynomial）、科依克变换（Koyck Transformation）等多种方法有目的地减少需要直接估计的模型参数个数，以缓解多重共线性，保证自由度。

（3）非平稳行为序列的自回归分布滞后模型的短期失衡问题

如果行动、效果序列独立来看是非平稳的，可通过差分的方法将其转化为平稳的，则其线性关系是平稳的，即在序列上进行水平上的回归，它们就可能存在协积关系或长期关系。但是，其在短期内可能会失衡。拟用误差修正模型（Error Correction Model，ECM）对短期失衡做出修正。

5.5 实现方案

为实现总体目标，解决关键问题，应结合聚焦爬虫（Focused Crawler）、自然语言处理（Natural Language Processing，NLP）、情感分析（Sentiment Analysis）、调整的热卡填充，基于 MapReduce 框架的分布式计算、线性及非线性回归、优化、滞后变量建模等关键技术展开研究。实现方案如图 5-2 所示。

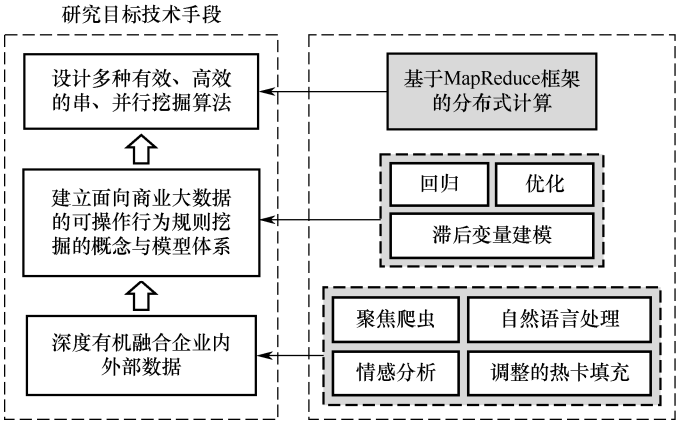


图 5-2　实现方案

5.5.1 采用的大数据技术

1. 聚焦爬虫

（1）网络爬虫的构成及分类

网络爬虫又称为网络蜘蛛、网络机器人，主要用于网络资源的收集工作。

在进行网络信息分析时，要获取网络信息内容，就需要用到网络爬虫这个工具，它是一个能自动提取网页内容的程序，通过搜索引擎（Search Engine），从互联网上爬取网页地址并抓取相应的网页内容，是搜索引擎的重要组成部分。

典型网络爬虫的主要组成部分如下：

① URL 链接库，主要用于存放爬取的网页链接。

② 文档内容模块，主要用于存取从 Web 中下载的网页内容。

③ 文档解析模块，用于解析下载文档中的网页内容，如解析 PDF、Word、HTML 等。

④ 存储文档的元数据及内容的库。

⑤ 规范化 URL 模块，用于把 URL 转换成标准的格式。

⑥ URL 过滤器，主要用于过滤不需要的 URL。

网络爬虫的主要工作是确定爬取的内容及爬取的范围。最简单的例子是从一个已知的站点抓取一些网页，这个爬虫用少量代码就可以完成。然而，在实际互联网应用中，可能会有爬取大量内容的需求，这时就需要设计一个较为复杂的基于分布式的爬虫。

（2）网络爬虫的工作原理

首先，选择初始 URL，并获得初始网页的域名或 IP 地址；然后，在抓取网页时，不断从当前页面上获取新的 URL 放入候选队列，直到满足停止条件。

聚焦爬虫（主题驱动爬虫）不同于传统爬虫，其工作流程比较复杂。首先，需要过滤与主题不相关的链接，只保留有用的链接并将其放入候选 URL 队列；然后，根据搜索策略从候选队列中选择下一个要抓取的网页链接，并重复上述过程，直到满足停止条件；与此同时，将所有爬取的网页内容保存起来，并进行过滤、分析、建立索引等，以便检索和查询。

总体来讲，网络爬虫主要有如下两个阶段：

第 1 阶段，URL 库初始化然后开始爬取。

第 2 阶段，爬虫通过读取没有访问过的 URL 来确定它的工作范围。

其中，对于所要爬取的 URL 链接，应进行以下步骤：

① 获取 URL 链接。

② 解析内容，获取 URL 及相关数据。

③ 存储有价值的数据。

④ 对新爬取的 URL 进行规范化。

⑤ 过滤不相关的 URL。

⑥ 将要爬取的 URL 更新到 URL 库中。

⑦ 重复步骤①，直到满足停止条件。

（3）网络爬虫的搜索策略

目前，比较常见的网络爬虫搜索策略有以下三种：

① 广度优先搜索。基本思想：首先访问根节点，若根节点有子节点，则访问所有子节点；然后依次广度优先遍历以所有子节点为根的子树。这种策略多用在主题爬虫上，因为越是与初始 URL 距离近的网页，其具有的主题相关性越大。

② 深度优先搜索。基本思想：首先访问根节点，若根节点有未被访问的子节点，则依次访问这些节点，同时深度优先遍历以当前访问节点为根的子树。

③ 最佳优先搜索。最佳优先搜索策略通过计算 URL 描述文本与目标网页的相似度，或者与主题的相关性，根据所设定的阈值选出有效 URL 进行抓取。

（4）爬行算法

数据采集的效率及覆盖率受爬行算法的影响，现在比较流行和经典的爬行

算法都是在 Best-Frist 算法的基础上改进和演化而来的。各种算法的不同之处在于对待爬行的 URL 采用不同的启发式规则进行打分并排序，同时在爬行之前或在爬行过程中对算法的参数进行优化。

① Best-First 算法。Best-First 算法通过维持一个排序的 URL 优先级队列，通过计算主题与所抓取网页的余弦相似度（Cosine Similarity）来确定 URL 队列（Frontier）中 URL 的优先级。

相似度计算公式如下：

$$\mathrm{sim}(q,p)=\frac{\sum_{k\in q\cap p} f_{kq}f_{kp}}{\sqrt{\sum_{k\in p} f_{kp}^2 \sum_{k\in q} f_{kq}^2}}$$

式中，q 为主题，p 为抓取的网页。

Best-Frist 爬行算法描述如下：

i. 初始化，设定查询主题（topic）、初始种子节点集合（starting_urls）、爬取的最大网页数量（MAX_PAGES）、frontier 的容量限制（MAX_BUFFER）。

ii. 把初始节点集合（starting_urls）中的所有 link 插入到 frontier（初始为空）。

iii. while（frontier 非空且爬行的网页数<MAX_PAGES）。

a）link=dequeue_link_with_max_score（frontier）

//从 frontier 中取出优先级最高的链接

b）doc=fetch_new_page（link）

//访问并下载该 link 对应的页面

c）score=sim（topic, doc）

//计算页面与主题的相似度

d）抽取 doc 中的所有超链接 outlinks

e）for doc 中的每个超链接 outlink

//完成 doc 中的链接入 frontier 的过程

if　#frontier>=MAX_BUFFER

```
        //当 frontier 中链接的数量大于等于最大限制
            dequeue_link_with_min_score（frontier）
            //从 frontier 删除最小优先级链接
        else
            enqueue（frontier, outlink, score）
            //把链接和优先级加入 frontier 队列
```

② Shark-Search 算法。Shark-Search 算法是 Hersovici 等人在 Fish-Search 算法的基础上进行一些改进得到的算法。具体来说，其使用矢量空间模型构建查询 q 与网页、链接锚文本及链接附近文本的空间矢量并计算其余弦相似度值（介于 0 和 1 之间的连续值），从而代替 Fish-Search 算法的关键字及正则表达式匹配的相关性为 0 或 1 的离散值。Shark-search 算法同时利用了锚文本、链接附近文本和父页面相关度信息三方面的相关度线索。同 Fish-Search 算法一样，该算法也设定了爬行的深度界限（Depth Bound）。在爬行队列中的每个链接都关联了一个深度界限和优先级度量值。深度界限由用户设定，链接的优先级值由父页面相关性、锚文本及链接附近文本共同决定。算法描述如下：

```
i.   初始化：设定初始节点，深度界限（D），爬行尺度（S），时限，搜索查询 q。
ii.  设定初始节点的深度界限 depth=D，并把节点插入到爬行队列（初始为空）。
iii. while（队列非空且下载的节点数<S，并且在规定的下载时限内）。
     a）if relevance(current_node)>0   //当前节点相关
          inherited_score(child_node)= δ *sim(q，current_node)
        //其中δ为预先设定的衰减因子=0.5
       else
          inherited_score(child_node)=δ*inherited_score(current_node)
       //计算 child_node，inherited_score(child_node)
     b）提取 anchor_text 及 anchor_text_context
```

//通过预先设定的边界，如 1

c）anchor_score=sim(q, anchor_text)

//计算锚文本的相关度值

d）if anchor_score >0

anchor_context_score = 1

else

anchor_context_score = sim(q，anchor_text_context)

//计算锚文本的相关度值

e）neihgorhood_score：$neihgorhood_{score}$=β*anchor_score+(1−β)*anchor_context_score，

//计算 anchor 的相关度。其中β为预设的常量等于 0.8

f） $potential_{score(child_{node})} = \gamma * inhertited_{score(child_{node})} + (1-\gamma) * neighborhood\_score(child\_node)$

//计算子节点的潜在值。其中，γ为预设的常量，小于 1，一般设为 0.5

2. 自然语言处理

自然语言处理技术是所有与自然语言的计算机处理有关的技术的统称，其目的是使计算机理解和接收人类用自然语言输入的指令，完成从一种语言到另一种语言的翻译功能。自然语言处理技术的研究，可以丰富计算机知识处理的研究内容，推动人工智能技术的发展。下面分析自然语言处理的关键技术。

（1）常用技术分类

① 模式匹配技术。模式匹配技术主要是计算机将输入的语言内容与其内已设定的单词模式与输入表达式相匹配的技术。例如，计算机的辅导答疑系统，当用户输入的问题在计算机的答疑库里找到相匹配的答案时，就会自动回答问题。但是，该技术不能一直保证用户输入的问题能得到相应的回答，于是很快对这种简单匹配式答疑系统进行了改进，即答疑库中增加了同义词和反义词，当用户输入关键词的同义词或反义词时，计算机同样能完成答疑，这种改进后的系统称为模糊匹配式答疑系统。

② 语法驱动的分析技术。语法驱动的分析技术是指通过语法规则，如词形词性、句子成分等规则，将输入的自然语言转化为相应的语法结构的技术。这种分析技术可分为上下文无关文法、转换文法、ATN 文法。上下文无关文法是最简单并且应用最为广泛的语法，其规则产生的语法分析树可以翻译大多数自然语言，但由于其处理的词句无关上下文，所以对于某些自然语言的分析是不合适的。转换文法克服了上下文无关文法中存在的一些缺点，能够利用转换规则重新安排分析树的结构，既能形成句子的表层结构，又能分析句子的深层结构。但其具有较大的不确定性。ATN 文法扩充了转移网络，相比其他语法加入了测试集合和寄存器，它比转移文法更能准确地分析输入的自然语言，但也具有复杂、脆弱、低效等缺点。

③ 语义文法。语义文法的分析原理与语法驱动相似，但其具有更大的优越性。语义文法中是对句子的语法和语义的共同分析，能够解决语法驱动分析中单一对语法分析带来的不足。它能够根据句子的语义，将输入的自然语言更通顺地表达出来，同时除去一些语法正确但不合语义的翻译。但是语义文法分析仍然有不容忽视的缺点，其分析的语句中有时会出现不合语法的现象，并且这类分析较为复杂，如语义类难以确定、语义的规则太多等。因此，语义文法技术仍需要改进。

④ 格框架约束分析技术。格框架是由一个头部和一组辅助概念组成的。头部一般由主要动词构成，辅助概念也称“域”，以某种规范形式与头部相连。格框架定义规定了与头部相应的必有格、随意格和禁止格。在进行格框架约束分析技术时，输入的自然语言被转化为格内容，它既结合了语法驱动分析技术和语义文法分析技术的优点，又克服了语义文法中不合文法的现象，解决了语句的多义性问题，是计算机语言研究中的重大发展之一。

⑤ 系统文法。系统文法是从多个层次分析自然语言的分析方法。它强调句子的整体结构，主要从语法、语义和语音等层次来分析自然语言。每个层次又有三种不同的分析，分别为功用说明、特征说明和组成成分结构分析。系统文

法可以根据自然语言的功能特性和组成成分来分析自然语言，但也有系统结构复杂等缺点。

⑥ 功能文法。功能文法是对句子的完全功能描述，它描述了自然语言的特征组合、功能分配、词语组成成分顺序，是一种既可以用于分析也可以用于生成的文法。功能文法的分析形式是分析自然语言的主动句规则、主谓一致规则，构成相应的字典入口形式。有一种与功能文法相似的文法系统为词功能文法，它更强调词典的功能。

⑦ 故事文法。显示计算机翻译输入的自然语言时，故事文法不仅从语句的语法、语义、结构的角度，还能够从整个故事的情节发展的角度将信息整合得准确到位。但此类文法一般只适用于处理较为简单的、文体较为形式化的故事描述，对于一些情节较为复杂的故事，不一定能精确描述。这种技术仍有待进一步发展研究。

（2）中文自然语言处理的关键技术

① 词法分析。词法分析包括词形和词汇两个方面。一般来讲，词形主要表现在对单词的前缀、后缀等的分析，而词汇则表现在对整个词汇系统的控制。在中文全文检索系统中，词法分析主要表现在对汉语信息进行词语切分上，即汉语自动分词技术。通过这种技术能够比较准确地分析用户输入信息的特征，从而完成准确的搜索过程。它是中文全文检索技术的重要发展方向。

② 句法分析。句法分析是对用户输入的自然语言进行词汇短语的分析，目的是识别句子的句法结构，实现自动句法分析过程。其基本方法有线图分析、短语结构分析、完全句法分析、局部句法分析、依存句法分析等。

③ 语义分析。语义分析是基于自然语言语义信息的一种分析方法，其不仅是如词法分析和句法分析等在语法水平上的分析，而且涉及单词、词组、句子、段落所包含的意义。其目的是根据句子的语义结构表示言语的结构。中文语义分析方法是基于语义网络的一种分析方法。语义网络则是一种结构化的，灵活、

明确、简洁的表达方式。

④ 语用分析。语用分析相对于语义分析又增加了对上下文、语言背景、环境等的分析，从文章的结构中提取意象、人际关系等附加信息，是一种更高级的语言学分析。它将语句中的内容与现实生活的细节相关联，从而形成动态的表意结构。

⑤ 语境分析。语境分析主要是指对原查询语篇之外的大量“空隙”进行分析，以更为正确地解释所要查询语言的技术。这些“空隙”包括一般的知识、特定领域的知识和查询用户的需要等。它将自然语言与客观的物理世界和主观的心理世界联系起来，补充完善了词法分析、语义分析、语用分析的不足。

3. 情感分析

随着企业信息化与互联网的发展，信息爆炸式增长，其中包括大量的非结构化与半结构化数据。非结构化与半结构化数据主要是文本型数据，阐述“5W”问题，即 Who、When、Where、What、Why。如何充分利用非结构化数据与半结构化数据，分析其包含的潜在信息以支持决策，成为众多企业与研究者关注的重点。其中，非常有价值的一类数据是互联网（如博客和论坛）上大量用户参与的，对人物、事件、产品等有价值的评论信息。这些评论信息表达了人们的各种情感色彩和情感倾向性，如喜、怒、哀、乐、批评、赞扬等。基于此，可以通过浏览这些主观色彩的评论来了解大众舆论对于某一事件或产品的看法。由于越来越多的用户乐于在互联网上分享自己的观点或体验，因此这类评论信息量迅速膨胀，仅靠人工的方法难以应对网上海量信息的收集和处理，因此迫切需要计算机帮助用户快速获取和整理这些相关评价信息。因此，如何从这些 Web 文本中进行情感挖掘，获取情感倾向，已经成为当今商务智能领域关注的热点。情感分析技术也由此应运而生。

文本情感分析又称为意见挖掘，是对带有情感色彩的主观性文本进行分析、处理、归纳和推理的过程。其中，主观情感可以是他们的判断或评价，他们的情绪状态，以及有意传递的情感信息。因此，情感分析的主要任务就是情感倾

向性的判断，Pang 等人将情感倾向分为正面、负面和中性，即褒义、贬义和客观评价。研究初期，大量研究者都致力于词语和句子的倾向性判断的研究，但随着互联网上大量主观性文本的出现，研究者们逐渐从简单的情感词语的分析研究过渡到更为复杂的情感句和情感篇章的研究。

文本情感分析可以归纳为 3 项层层递进的研究任务，即情感信息的抽取、情感信息的分类，以及情感信息的检索与归纳。情感信息的抽取就是将无结构的情感文本转化为计算机容易识别和处理的结构化文本。情感信息的分类则是利用情感信息抽取的结果将情感文本单元分为若干类别，供用户查看，如分为褒、贬、客观，或者其他更细致的情感类别。情感信息检索和归纳可以看作是与用户直接交互的接口，强调检索和归纳的两项应用。

情感分析是一个新兴的研究课题，具有很大的研究价值和应用价值，受到国内外众多研究者的青睐。目前，实现情感分析的技术主要包括基于机器学习的方法和基于语义的方法两类。

（1）基于机器学习的方法

随着大规模语料库的建设和各种语言知识库的出现，基于语料库的统计机器学习方法进入自然语言处理的视野。多种机器学习方法应用到自然语言处理中并取得了良好的效果，促进了自然语言处理技术的发展。机器学习的本质是基于数据的学习（Learning from Data）。利用机器学习算法可对统计语言模型进行训练，最后用训练好的分类器对新文本情感进行识别。

近年来，有关自然语言处理、人工智能、信息检索、数据挖掘和 Web 应用等领域的多个国际顶级会议（AAAI、ACL、SIGIR 等）都收录了文本情感倾向分析的相关论文。

目前，虽然机器学习方法的分类准确度比较高，但是训练样本集的建立需要采用人工方法对大量的评论文章进行逐一阅读甄别，并进行手工标引，这与利用自动情感分类降低人的阅读负担这一初衷还有着一定的差距。因此，近来许多研究者将情感分析研究的重点集中在对训练样本的需求量较低的语

义方法上。

（2）基于语义的方法

有些学者曾经想利用词典将手工采集的种子评价词语进行扩展，来获取大量的评价词。这种方法简单易行，但是较依赖种子评价词语的个数和质量，并且容易由于一些词语的多义性而引入噪声。为了避免词语的多义性，一部分学者使用词典中词语的注释信息来完成评价词语的识别与极性判断。此外，一些学者沿用了 Turney 等人的点互信息的方法，通过计算 WordNet 中的所有形容词与种子褒义词代表 good 和贬义词代表 bad 之间的关联度值来识别评价词语的情感倾向。

4. MapReduce

（1）Hadoop 与 MapReduce

Hadoop 是一个分布式系统基础架构，由 Apache 基金会开发。用户可以在不了解分布式底层细节的情况下开发分布式程序，充分利用集群的威力进行高速运算和存储。

Hadoop 由 Pig、Chukwa、Hive、HBase、MapReduce、HDFS、ZooKeeper、Core、Avro 九部分组成，最核心的设计是 MapReduce 和 HDFS。Hadoop 部分子项目的作用如表 5-1 所示。

表 5-1　Hadoop 部分子项目的作用

子项目	作用
Pig	一种用于探索大型数据集的脚本语言
Chukwa	展示、监控和分析已收集的数据
Hive	提供类似 Oracle 的数据添加、查询、修改、删除方法
Hbase	提供可靠的、可扩展的分布式数据库
Zookeeper	为分布式提供高一致性服务
Core	提供了一个分布式文件系统（HDFS）和支持 MapReduce 的分布式计算
Avro	序列化，提高分布式传输效率

MapReduce 是一种编程模型，用于大规模数据集（大于 1TB）的并行运算。

概念“Map（映射）”和“Reduce（归约）”是它们的主要思想，都是从函数式编程语言里借来的，还有从矢量编程语言里借来的特性。它极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上。 当前的软件实现是指定一个 Map 函数，用来把一组键值对映射成一组新的键值对，指定并发的 Reduce 函数，用来保证所有映射的键值对中的每个都共享相同的键组。

（2）MapReduce 编程模型

MapReduce 采用“分而治之”的思想，把对大规模数据集的操作，分发给一个主节点管理下的各个分节点共同完成，然后通过整合各个节点的中间结果，得到最终结果。简单地说，MapReduce 就是“任务的分解与结果的汇总”。

在 Hadoop 中，用于执行 MapReduce 任务的机器角色有两个：一个是 JobTracker；另一个是 TaskTracker。JobTracker 用于调度工作，TaskTracker 用于执行工作。一个 Hadoop 集群中只有一台 JobTracker。

在分布式计算中，MapReduce 框架负责处理并行编程中分布式存储、工作调度、负载均衡、容错均衡、容错处理、网络通信等复杂问题。其处理过程可高度抽象为 map 和 reduce 两个函数，map 负责把任务分解成多个任务，reduce 负责把分解后多任务处理的结果汇总起来。

需要注意的是，用 MapReduce 来处理的数据集（或任务）必须具备这样的特点：待处理的数据集可以分解成许多小的数据集，而且每个小数据集都可以完全并行地进行处理。

（3）MapReduce 处理过程

在 Hadoop 中，每个 MapReduce 任务都被初始化为一个 Job，每个 Job 又可以分为两种阶段：Map 阶段和 Reduce 阶段。这两个阶段分别用两个函数表示，即 map 函数和 reduce 函数。map 函数接收一个<key, value>形式的输入，然后同样产生一个<key, value>形式的中间输出，Hadoop 函数接收一个如<key,

(list of values)>形式的输入，然后对这个 value 集合进行处理，每个 reduce 产生 0 或 1 个输出，reduce 的输出也是<key, value>形式的。

MapReduce 处理过程如图 5-3 所示。一切都是从最上方的 User Program 开始的，User Program 链接了 MapReduce 库，实现了最基本的 map 函数和 reduce 函数。图中执行的顺序都用数字进行了标记。

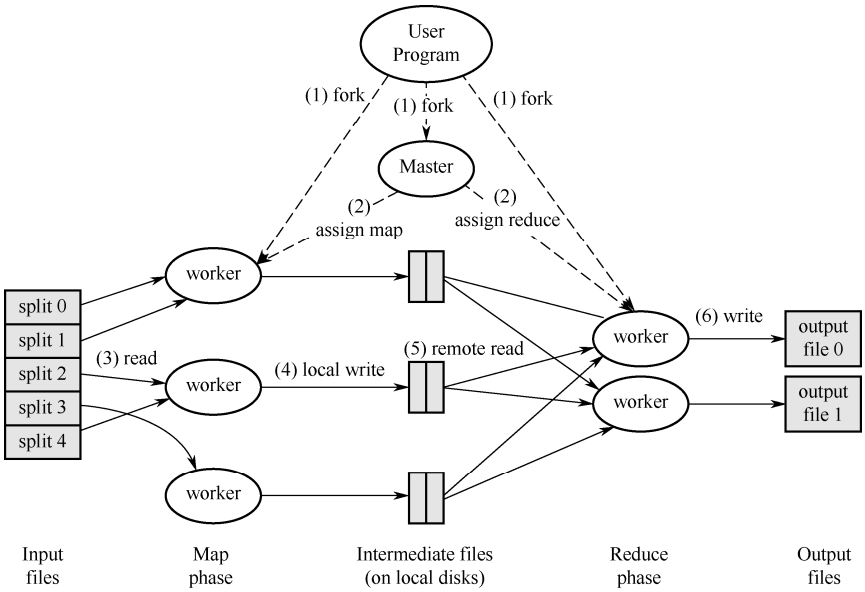


图 5-3　MapReduce 处理过程

① MapReduce 库先把 User Program 的输入文件划分为 *M* 份（*M* 为用户定义），每份通常有 16～64MB，如图 5-3 左侧所示分成了 split 0～4；然后使用 fork 将用户进程复制到集群内其他机器上。

② User Program 的副本中有一个称为 Master，其余称为 worker，Master 是负责调度的，为空闲 worker 分配作业（map 作业或者 reduce 作业），worker 的数量也是可以由用户指定的。

③ 被分配了 map 作业的 worker，开始读取对应分片的输入数据，map 作业数量是由 *M* 决定的，和 split 一一对应；map 作业从输入数据中抽取出键值

对，每个键值对都作为参数传递给 map 函数，map 函数产生的中间键值对被缓存在内存中。

④ 缓存的中间键值对会被定期写入本地磁盘，而且被分为 R 个区，R 的大小是由用户定义的，将来每个区会对应一个 reduce 作业；这些中间键值对的位置会被通报给 Master，Master 负责将信息转发给 reduce worker。

⑤ Master 通知分配了 reduce 作业的 worker 其负责的分区在什么位置（位置有多个，每个 Map 作业产生的中间键值对都可能映射到所有 R 个不同分区），当 reduce worker 把所有由其负责的中间键值对都读取后，先对它们进行排序，使得相同键的键值对聚集在一起。因为不同的键可能会映射到同一个分区，也就是同一个 reduce 作业，所以排序是必须的。

⑥ reduce worker 遍历排序后的中间键值对，对于每个唯一的键，都将键与关联的值传递给 reduce 函数，reduce 函数产生的输出会添加到这个分区的输出文件中。

⑦ 当所有的 map 和 reduce 作业都完成后，Master 唤醒 User Program，map/reduce 函数调用返回 User Program 的代码。

所有过程执行完毕后，MapReduce 的输出放在了 R 个分区的输出文件中（分别对应一个 reduce 作业）。用户通常并不需要合并这 R 个文件，而是将其作为输入交给另一个 MapReduce 程序处理。整个过程中，输入数据来自底层分布式文件系统（GFS），中间数据放在本地文件系统，最终输出数据写入底层分布式文件系统（GFS）。要注意 map/reduce 作业和 map/reduce 函数的区别：map 作业处理一个输入数据的分片，可能需要调用多次 map 函数来处理每个输入键值对；reduce 作业处理一个分区的中间键值对，期间要对每个不同的键调用一次 reduce 函数，reduce 作业最终也对应一个输出文件。

（4）MapReduce 作业运行流程

MapReduce 作业运行流程如图 5-4 所示。

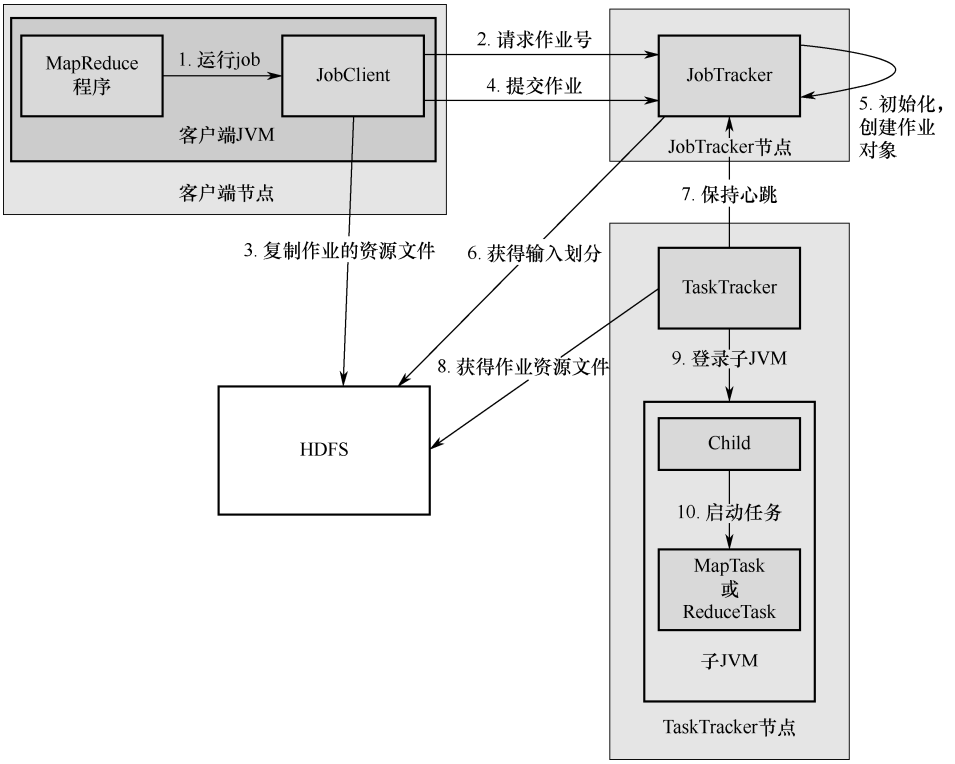


图 5-4　MapReduce 作业运行流程

流程分析如下：

① 在客户端启动一个作业。

② 向 JobTracker 请求一个 Job ID。

③ 将运行作业所需要的资源文件复制到 HDFS 上，包括 MapReduce 程序打包的 JAR 文件、配置文件和客户端计算所得的输入划分信息。这些文件都存放在 JobTracker 专门为该作业创建的文件夹中。文件夹名为该作业的 Job ID。JAR 文件默认有 10 个副本（mapred.submit.replication 属性控制）；输入划分信息告诉 JobTracker 应该为这个作业启动多少个 map 任务等信息。

④ JobTracker 接收到作业后，将其放在一个作业队列里，等待作业调度器对其进行调度，当作业调度器根据自己的调度算法调度到该作业时，会根据输

入划分信息为每个划分创建一个 map 任务，并将 map 任务分配给 TaskTracker 执行。对于 map 和 reduce 任务，TaskTracker 根据主机核的数量和内存的大小分配固定数量的 map 槽和 reduce 槽。map 任务通过数据本地化（Data-Local）模式分配给某个 TaskTracker：将 map 任务分配给含有该 map 处理的数据块的 TaskTracker 上，同时将程序 JAR 包复制到该 TaskTracker 上来运行（称为“运算移动，数据不移动”），而分配 reduce 任务时并不考虑数据本地化。

⑤ TaskTracker 每隔一段时间会给 JobTracker 发送一个心跳，告诉 JobTracker 它依然在运行，同时心跳中还携带着很多信息，如当前 map 任务完成的进度等。当 JobTracker 收到作业的最后一个任务完成信息时，把该作业设置为“成功”。当 JobClient 查询状态时，它得知任务已完成，便会显示一条消息给用户。

以上是在客户端、JobTracker、TaskTracker 的层次来分析 MapReduce 的工作原理，下面从 map 任务和 reduce 任务的层次来分析。

（5）Map、Reduce 任务中 Shuffle 和排序的过程

Map、Reduce 任务中 Shuffle 和排序的过程如图 5-5 所示。

下面是 map 端流程分析：

① 每个输入分片会让一个 map 任务来处理，在默认情况下，以 HDFS 的一个块的大小（默认为 64MB）为一个分片，当然也可以设置块的大小。map 输出的结果会暂且放在一个环形内存缓冲区中（该缓冲区的大小默认为 100MB，由 io.sort.mb 属性控制），当该缓冲区快要溢出时（默认为缓冲区大小的 80%，由 io.sort.spill.percent 属性控制），会在本地文件系统中创建一个溢出文件，将该缓冲区中的数据写入这个文件。

② 在写入磁盘之前，线程首先根据 reduce 任务的数目将数据划分为相同数目的分区，也就是一个 reduce 任务对应一个分区的数据。这样做是为了避免有些 reduce 任务分到大量数据，而有些 reduce 任务只能分到很少甚至没有分到

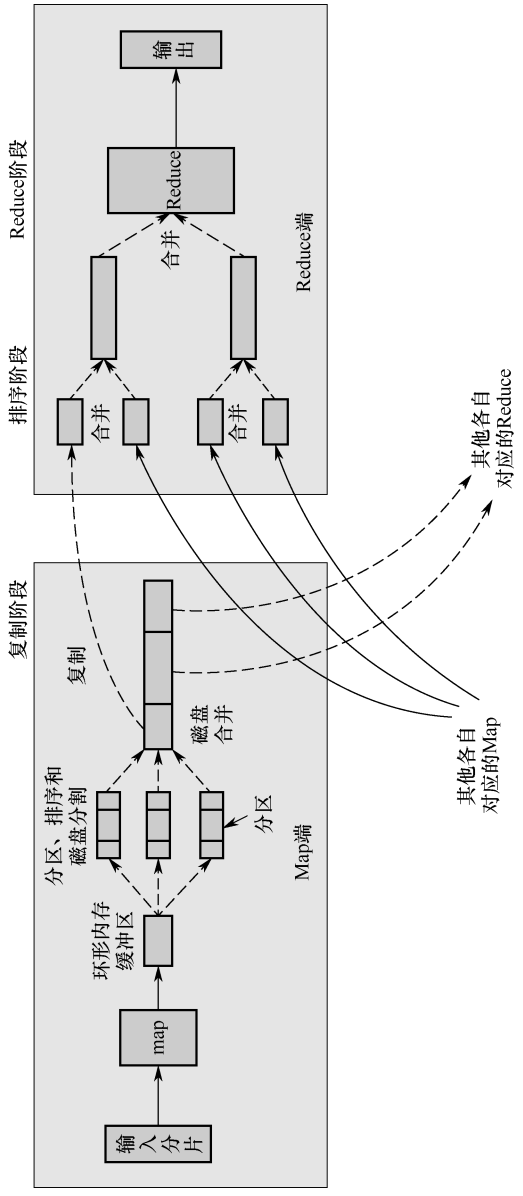


图 5-5　Map、Reduce 任务中 Shuffle 和排序的过程

数据的尴尬局面。其实分区就是对数据进行 hash 的过程。然后对每个分区中的数据进行排序，如果此时设置了 Combiner，则对排序后的结果进行 Combiner 操作，这样做的目的是让尽可能少的数据写入磁盘。

③ 当 map 任务输出最后一个记录时，可能会有很多溢出文件，这时需要将这些文件合并。合并的过程中会不断地进行排序和 Combiner 操作，目的有两个：第一，尽量减少每次写入磁盘的数据量；第二，尽量减少下一复制阶段网络传输的数据量。最后合并成一个已分区且已排序的文件。为了减少网络传输的数据量，这里可以将数据压缩，只要将 mapred.compress.map.out 设置为 true 就可以了。

④ 将分区中的数据复制给相对应的 reduce 任务。map 任务一直和其父 TaskTracker 保持联系，而 TaskTracker 又一直和 JobTracker 保持心跳。所以 JobTracker 中保存了整个集群中的宏观信息，只要 reduce 任务向 JobTracker 获取对应的 map 输出位置就可以了。

Shuffle 的中文意思是“洗牌”：一个 map 产生的数据，结果却通过 hash 过程分区分配给了不同的 reduce 任务。

下面是 reduce 端流程分析：

① reduce 会收到不同 map 任务传来的数据，并且每个 map 传来的数据都是有序的。如果 reduce 端接收的数据量相当小，则直接存储在内存中（缓冲区大小由 mapred.job.shuffle.input.buffer.percent 属性控制，表示用作此用途的堆空间的百分比），如果数据量超过了该缓冲区大小的一定比例（由 mapred.job.shuffle.merge.percent 决定），则将数据合并后溢写到磁盘中。

② 随着溢写文件的增多，后台线程会将它们合并成一个更大的有序的文件，这样做是为了给后面的合并节省时间。其实不管在 map 端还是 reduce 端，MapReduce 都是反复地执行排序、合并操作。

③ 合并的过程中会产生许多中间文件（被写入磁盘），但 MapReduce 会让

写入磁盘的数据尽可能地少，并且最后一次合并的结果并没有写入磁盘，而是直接输入 reduce 函数。

（6）物理实体

可以从很多不同的角度描述 MapReduce 运行机制，如从 MapReduce 运行流程和计算模型的逻辑流程来描述，也许在深入理解 MapReduce 运行机制后还会从更好的角度来描述。从任何角度描述的 MapReduce 都必须包含参入的实例对象与计算模型的逻辑定义阶段。

参入 map/reduce 作业执行涉及 4 个独立的实体：

① 客户端（Client）：编写 map/reduce 程序，配置作业，提交作业，由程序员完成。

② JobTracker：初始化作业，分配作业，与 TaskTracker 通信，协调整个作业的执行。

③ TaskTracker：保持与 JobTracker 的通信，在分配的数据片段上执行 map 或 reduce 任务。TaskTracker 和 JobTracker 的重要区别是，在执行任务的时候，TaskTracker 可以有 *n* 个，而 JobTracker 只有一个。

④ HDFS：保存作业的数据、配置信息等，最后的结果也保存在 HDFS 上。

（7）运行原理

MapReduce 运行原理如图 5-6 所示。

① 客户端编写 map/reduce 程序，配置 map/reduce 的作业即 job。

② 提交 job 到 JobTracker 上。这时候 JobTracker 会构建这个 job，具体就是分配一个新的 job 任务的 ID 值。

③ JobTracker 做检查操作（确定输出目录与输入目录是否存在）。如果输出目录不存在，那么 job 就不能正常运行下去，JobTracker 会抛出错误信息给客户端。如果输入目录不存在，同样抛出错误信息，否则 JobTracker 会根据输入计

算输入分片（Input Split）。如果分片计算不出来，也会抛出错误信息。

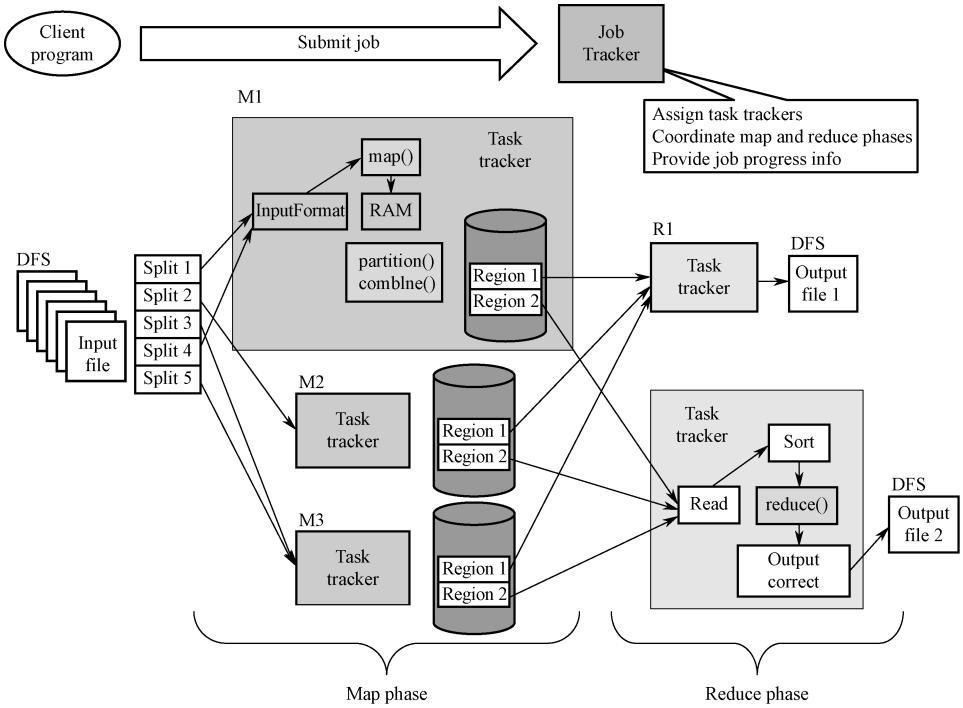


图 5-6　MapReduce 运行原理

④ JobTracker 配置 job 需要的资源。

⑤ JobTracker 初始化作业。初始化的主要任务就是将 job 放入一个内部的队列，让配置好的作业调度器能调度到这个作业。作业调度器会初始化这个 job：创建一个正在运行的 job 对象（封装任务和记录信息），以便 JobTracker 跟踪 job 的状态和进程。

⑥ 作业调度器获取输入分片信息（Input Split），每个分片创建一个 map 任务。

⑦ 任务分配。TaskTracker 会运行一个简单的循环机制，定期发送心跳给 JobTracker，心跳间隔是 5s，程序员可以配置这个时间。心跳就是 JobTracker 和 TaskTracker 沟通的桥梁。通过心跳，JobTracker 可以监控 TaskTracker 是否

存活，也可以获取 TaskTracker 处理的状态和问题；同时，TaskTracker 可以通过心跳里的返回值获取 JobTracker 给它的操作指令。

⑧ 执行任务。在任务执行时，JobTracker 可以通过心跳机制监控 TaskTracker 的状态和进度，同时也能计算整个 job 的状态和进度，而 TaskTracker 可以本地监控自己的状态和进度。当 JobTracker 获得最后一个完成指定任务的 TaskTracker 操作成功的通知时，会把整个 job 状态置为成功。客户端查询 job 运行状态时（异步操作），会查到 job 完成的通知。如果中途失败，则 MapReduce 有相应的处理机制。一般而言，只要不是程序本身有 bug，MapReduce 错误处理机制都能保证提交的 job 能正常完成。

下面从逻辑实体的角度描述 MapReduce 运行机制（见图 5-7）。其按照时间顺序包括输入分片（Input Split）、map 阶段、combiner 阶段、shuffle 阶段和 reduce 阶段。

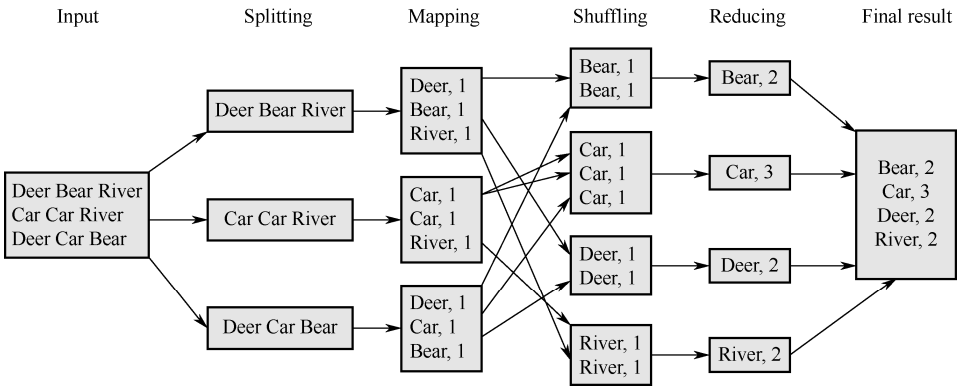


图 5-7　MapReduce 运行机制（逻辑实体角度）

① Input Split：在进行 map 计算之前，MapReduce 会根据输入文件计算 Input Split，每个 Input Split 针对一个 map 任务。Input Split 存储的并非数据本身，而是一个分片长度和一个记录数据的位置的数组。Input Split 往往与 HDFS 的 block（块）关系很密切。假如设定 HDFS 的块的大小是 64MB，输入有三个文件，大小分别是 3MB、65MB 和 127MB，那么 MapReduce 会把 3MB 文件分为

一个 Input Split，65MB 文件分为两个 Input Split，而 127MB 也分为两个 Input Split。换句话说，如果在 map 计算前做输入分片调整，如合并小文件，那么就会有 5 个 map 任务将执行，而且每个 map 执行的数据大小不均，这个也是 MapReduce 优化计算的一个关键点。

② map 阶段：程序员编写好 map 函数。一般 map 操作都是本地化操作，也就是在数据存储节点上进行。

③ combiner 阶段：combiner 阶段是程序员可以选择的，它也是一种 reduce 操作，因此，在 WordCount 类里用 reduce 进行加载。combiner 是一个本地化的 reduce 操作，它是 map 运算的后续操作，主要是在 map 计算出中间文件前做一个简单的合并重复 key 值的操作。例如，对文件里的单词频率做统计，在 map 计算时碰到一个“Hadoop”单词就会记录为 1，如果文章中出现 *n* 次“Hadoop”，那么 map 输出文件冗余就会很多。因此，在 reduce 计算前对相同的 key 做一个合并操作，文件就会变小，这样就提高了带宽的传输效率，毕竟带宽问题是 Hadoop 的主要计算瓶颈之一，也是最为宝贵的资源。但是 combiner 操作是有风险的，使用它的原则是 combiner 的输入不会影响 reduce 计算的最终输入。例如，如果计算只是求总数、最大值、最小值，则可以使用 combiner，但是如果使用 combiner 做平均值计算，则最终的 reduce 计算结果会出错。

④ shuffle 阶段：将 map 的输出作为 reduce 的输入的过程就是 shuffle，它是 MapReduce 优化的重点部分。shuffle 一开始就是 map 阶段做输出的操作。MapReduce 计算的一般都是海量数据，map 输出时不可能把所有文件都放到内存操作，因此 map 写入磁盘的过程十分复杂，更何况 map 输出时要对结果进行排序，需要的内存很大。map 在做输出时会在内存里开启一个环形内存缓冲区，这个缓冲区专门用来输出，默认大小是 100MB，并且在配置文件里为这个缓冲区设定了一个阈值，默认为 0.80（大小和阈值都可以在配置文件里进行配置）。同时 map 还会为输出操作启动一个守护线程，如果缓冲区的

内存达到阈值的 80%，则这个守护线程会把内容写到磁盘上（这个过程称为 spill）。另外的 20%内存可以用来继续写入要写进磁盘的数据。写入磁盘和写入内存操作是互不干扰的，如果缓存区被撑满，那么 map 就会阻止写入内存的操作，让写入磁盘操作完成后再继续执行写入内存操作。数据在写入磁盘前会有一个排序操作，如果定义了 combiner 函数，那么排序前还会执行 combiner 操作。每次 spill 操作（也就是写入磁盘操作）都会写一个溢出文件，也就是说，在做 map 输出时，有几次 spill 就会产生多少个溢出文件。map 输出全部完成后，map 会合并这些输出文件。在这个过程中还会有一个 partitioner 操作，其与 map 阶段的 input split 类似。一个 partitioner 对应一个 reduce 作业。因此，partitioner 就是 reduce 的输入分片，可以由程序员根据实际 key 和 value 的值、实际业务类型或者更好的 reduce 负载均衡的要求进行编程控制。这是提高 reduce 效率的关键。在 reduce 阶段，partitioner 会找到对应的 map 输出文件，然后进行复制操作。复制操作时，reduce 会开启几个复制线程，这些线程默认数量是 5 个。程序员也可以在配置文件中更改复制线程的数量。这个复制过程和 map 写入磁盘过程类似，也有阈值和内存大小，阈值一样可以在配置文件里配置，而内存大小是直接使用 reduce 的 TaskTracker 的内存大小。复制时，reduce 会进行排序操作和合并文件操作，之后就会进行 reduce 计算。

⑤ reduce 阶段：由程序员编写，最终结果存储在 HDFS。

5.5.2 企业内外部数据融合

为实现内外部数据的深度有机融合，拟采取以下所述技术策略与路线。

1. 互联网源数据获取

采用聚焦爬虫技术从海量互联网网页、论坛、博客、新闻、微博、社交媒

体等来源获取关于企业及其产品、服务，竞争对手，产业政策，宏观微观经济数据等原始数据。

2. 源数据预处理

Web 上抓取的网页中存在大量用户并不关心的信息，如导航条、广告信息、版权信息、调查问卷等内容，这些信息称为“网页噪声”。网页噪声导致主题漂移，使同一网页存在多个主题的情况。以整个网页为粒度的信息搜索结果不够准确，必须深入网页内部，找出网页主题，才能提高信息检索的准确性。

3. 虚假信息过滤

采用异常相关发表时间模式、发帖者关系网络分析、文本评论相似度等方法过滤虚假商品评价等信息。

4. 源数据的大规模存储

基于 Lucene 设计时间和空间复杂度良好的大规模数据索引方案，实现跨设备和数据中心存储，利用数据块技术将数据保存在物理上互不相关的多个磁盘中。

5. 数据提取

首先，使用自然语言理解技术，通过词法分析、句法分析、观点词选择、相似度计算、极性分析，定位观点词，确定观点词在句子中的语义倾向性。然后，使用情感分析等技术获取感兴趣的外部数据项样本。

6. 数据分片

按照内部行为数据集的时间间隔对所获得的外部数据项样本进行分片操作。

7. 增强数据可靠性

为每个外部数据属性设定可靠性阈值，对其来源样本数目小于该阈值的数

据项执行以下操作：

① 采用热卡填充法计算该数据项在空缺状态下的填充值 V_1。

② 假设该数据项原值为 V_0，按以下公式计算该数据项的修正值：

$$V' = V_0 N / T + V_1(1 - N / T)$$

其中，N、T 分别表示该数据项的来源样本数目与其所属属性的可靠性阈值。

8. 数据合并

把属于同一时间片的内外部数据合并为一条行为样本，从而获得融合内外部数据的全景式行为数据集。

5.5.3 模型构建

1. 规则建模

（1）基于线性回归的规则建模

在统计学中，线性回归是利用线性回归方程的最小平方函数对一个或多个自变量和因变量之间的关系进行建模的一种回归分析。线性回归模型简单、应用广泛。基于线性回归的规则模型可以作为其他规则模型的比较基准。

假定 I 为一个 BIS，则基于线性回归的规则模型可表示为

$$r_{LR} = (\{t_i \mid 1 \leqslant i \leqslant |A_{en}|\}, \{e_j \mid 1 \leqslant j \leqslant |A_{be}|\})$$

其中，$\boldsymbol{y} = ((\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{B})\boldsymbol{x}$

$$\boldsymbol{E} = \begin{bmatrix} 1 & \rho(o_1, t_1.a) & \rho(o_1, t_2.a) & \cdots & \rho(o_1, t_{|S|}.a) \\ 1 & \rho(o_2, t_1.a) & \rho(o_2, t_2.a) & \cdots & \rho(o_2, t_{|S|}.a) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \rho(o_{|o|}, t_1.a) & \rho(o_{|o|}, t_2.a) & \cdots & \rho(o_{|o|}, t_{|S|}.a) \end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix}\rho(o_1,e_1.a) & \rho(o_1,e_2.a) & \cdots & \rho(o_1,e_{|C|}.a)\\ \rho(o_2,e_1.a) & \rho(o_2,e_2.a) & \cdots & \rho(o_2,e_{|C|}.a)\\ \vdots & \vdots & & \vdots\\ \rho(o_{|o|},e_1.a) & \rho(o_{|o|},e_2.a) & \cdots & \rho(o_{|o|},e_{|C|}.a)\end{bmatrix},o_k\in O,1\leqslant k\leqslant|O|$$

$$\boldsymbol{x}=[1 \quad t_{1.}vt \quad t_{2.}vt \quad \cdots \quad t_{|A_{en}|.}vt]^{\mathrm{T}}$$

$$\boldsymbol{y}=[e_{1.}vt \quad e_{2.}vt \quad \cdots \quad e_{|A_{be}|.}vt]^{\mathrm{T}}$$

（2）基于支持矢量回归的规则建模

为适应训练样本集的非线性，传统的拟合方法通常在线性方程后面加入高阶项，但由此增加的可调参数会增加过拟合的风险。而支持矢量回归采用核函数代替线性方程中的线性项，可以使原来的线性算法“非线性化”，即能做非线性回归。与此同时，引进核函数达到了“升维”的目的，而增加的可调参数使得过拟合依然能控制。

假定 I 为一个 BIS，则基于支持矢量回归的规则模型可表示为

$$r_{SVR}=(\{t_i\mid 1\leqslant i\leqslant|A_{en}|\},\{e_j\mid 1\leqslant j\leqslant|A_{be}|\})$$

其中，$e_{i.}vt=\sum_{j=1}^{|O|}\alpha_j\rho(o_j,e_i,a)\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_j)^2}{2\sigma^2}\right)+\beta$，$\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_j)^2}{2\sigma^2}\right)$ 为核函数，且

$$\boldsymbol{x}=[t_{1.}vt \quad t_{2.}vt \quad \cdots \quad t_{|A_{en}|.}vt]^{\mathrm{T}}$$

$$\boldsymbol{x}_j=[\rho(o_j,t_{1.}a) \quad \rho(o_j,t_{2.}a) \quad \cdots \quad \rho(o_j,t_{|A_{en}|.}a)]^{\mathrm{T}}$$

$$\boldsymbol{x}=[t_{1.}vt \quad t_{2.}vt \quad \cdots \quad t_{|A_{en}|.}vt]^{\mathrm{T}}$$

（3）基于高斯过程回归的规则建模

高斯过程回归是近年发展起来的一种机器学习回归方法，其有严格的统计学理论基础，对处理高维数、非线性等复杂的问题具有很好的适应性，且泛化能力强，具有容易实现、超参数自适应获取、非参数推断灵活、输出具有概率

意义等优点。

假定 I 为一个 BIS，则基于高斯过程回归的规则模型可表示为

$$r_{GPR}=(\{t_i\mid 1\leqslant i\leqslant|A_{en}|\},\{e_j\mid 1\leqslant j\leqslant|A_{be}|\})$$

其中，$e_{i.}vt=\boldsymbol{K}_*\boldsymbol{D}'\boldsymbol{y}_i$，$\boldsymbol{D}=\boldsymbol{K}+\sigma_n^2\boldsymbol{I}_n$

式中

$$n=|O|,\quad \boldsymbol{K}_*=[k(\boldsymbol{x},\boldsymbol{x}_1)\quad k(\boldsymbol{x},\boldsymbol{x}_2)\quad\cdots\quad k(\boldsymbol{x},\boldsymbol{x}_{|O|})]$$

其中，

$$\boldsymbol{x}_p=[\rho(o_p,t_{1.}a)\quad\rho(o_p,t_{2.}a)\quad\cdots\quad\rho(o_p,t_{|A_{en}|.}a)]^{\mathrm{T}}$$

$$\boldsymbol{y}_i=[\rho(o_1,e_{i.}a)\quad\rho(o_2,e_{i.}a)\quad\cdots\quad\rho(o_{|o|},e_{i.}a)]^{\mathrm{T}}$$

$k(\boldsymbol{\alpha},\boldsymbol{\beta})$ 表示矢量$\boldsymbol{\alpha}$和$\boldsymbol{\beta}$的协方差，$\boldsymbol{K}=(k_{pq})_{n\times n}$，$\sigma_n^2$ 是噪声方差。

（4）效果的滞后效应建模

① 分布滞后模型。如果模型中的滞后变量只包含行动，也就是说，模型中包含行动的当期值和若干期滞后值，则称该模型为分布滞后模型。

首先，通过相关系数、调整的判定系数，以及施瓦茨准则 SC 等统计检验确定滞后期长度。然后，采用阿尔蒙变换，用滞后期 i 的适当次多项式来逼近回归系数 β_i。变换后的模型变量数目将显著少于初始模型，从而达到降低多重共线性的目的。最后，对变换后的模型进行 OLS 估计，得到规则的分布滞后模型。

② 自回归模型。如果模型中的滞后变量只包含效果，也就是说，模型中包含行动的当期值和效果的若干期滞后值，则称该模型为自回归模型。

首先，通过相关系数、调整的判定系数，以及施瓦茨准则 SC 等统计检验确定滞后期长度。然后，建立初始自回归模型，采用广义差分法（Generalized Finite Difference Method）修正回归模型中的残差序列相关性。最后，建立修正

后的规则的自回归模型。

③ 自回归分布滞后模型。如果模型中的滞后变量既包括行动又包括效果，也就是说，模型中包含行动的当期值和若干期滞后值，以及效果的若干期滞后值，则称该模型为自回归分布滞后模型。

首先，通过相关系数、调整的判定系数，以及施瓦茨准则 SC 等统计检验确定滞后期长度。然后，对行动、效果变量进行协整分析，以发现变量之间的协整关系，即长期均衡关系，并以这种关系构成误差修正项。最后，建立短期模型，将误差修正项看作一个行动，连同其他反映短期波动的行动一起，建立规则的自回归分布滞后模型，即误差修正模型。

5.5.4　算法设计

1. 串行挖掘算法

基于前述各种规则模型可设计多种串行挖掘算法。首先，根据规则模型与 EU 的定义，可得到相应的规则效用模型。然后，用最优化方法求得具有最大 EU 的 ABR。

2. 并行挖掘算法

随着云计算的兴起，MapReduce 框架因具有能隐藏数据分配、容错、负载均衡等问题而使用户只需专注算法本身等显著优点，成为最成功的云计算框架之一。因此，可基于 MapReduce 框架为各种串行挖掘算法设计并行化方案。拟采用的基本规程如下：

（1）数据划分和计算任务调度

将一个挖掘任务（job）待处理的大数据划分为多个数据块，每个数据块对应一个计算任务（task），并利用 MapReduce 自动调度计算节点来处理相应的数据块。

（2）数据/代码互定位

为了减少数据通信，一个基本原则是本地化数据处理，即一个计算节点尽可能处理其本地磁盘上所分布存储的数据，这实现了代码向数据的迁移。

（3）系统优化

为了减少数据通信开销，中间结果数据在进入 reduce 节点前，进行一定的合并处理；一个 reduce 节点所处理的数据可能来自多个 map 节点。

第 6 章

总　结

本书紧紧围绕组织行为模式挖掘问题进行了深入研究，包括提高组织行为预测模型的质量和建立一类新的组织行为模式挖掘问题——可操作行为规则挖掘。其中，可操作行为规则挖掘也可应用于国家、群体等其他实体的行为建模，在国家安全、公共政策、商务智能等领域有着广泛的应用前景。

本书的主要阐述了有以下几方面：

① 使用三种评价指标比较分析了主要的分类方法，包括朴素贝叶斯、支持矢量机、人工神经网络、k-最近邻、决策树、随机森林、关联分类，所建立的组织行为预测模型的性能，为不同情形下分类方法的恰当选择提供了依据。实验结果表明，朴素贝叶斯在各种指标下都有最优的性能。然而，所有分类器的召回率都较低，而 AUC 值依分类器不同而变化较大。这是因为类不平衡问题严重阻碍了标准分类器的性能。

② 针对本领域普遍存在的类不平衡与非一致误分类代价问题，本书研究了四种典型代价敏感学习方法基于不同标准分类器建立的预测模型在不同情形下的性能。研究证实了代价敏感学习方法在本领域的有效性并建议了不同情形下对代价敏感学习方法、基分类器，以及方法-分类器组合的选择。首先，上采样方法是解决组织行为预测建模中类不平衡与非一致误分类代价问题的较优代价敏感学习方法。另外，总体上，MLP、NB 和 RF 都是基分类器的较好选择，而在类高度不平衡条件下，NB 是最好的选择。尽管大部分方法-分类器组合总体上都是有效的，但高的类不平衡程度将阻碍代价敏感学习方法的性能。

③ 为避免上采样方法建立的组织行为预测模型可能存在的过拟合问题，提出了一个对不同正样本采用不同复制策略的代价敏感算法 OESP。实验证实其在组织行为预测建模领域比上采样等代价敏感学习方法更有效。

④ 基于代价曲线提出了一个针对组织行为预测建模中类不平衡与非一致误分类代价问题的有效个性化解决方案。该方案可使用户方便、直观地为给定数据集选择最优代价敏感学习方法-分类器组合。

⑤ 建立了一类新的组织行为模式挖掘问题——可操作行为规则挖掘。首

先，提出了可操作行为规则挖掘问题的形式化定义。然后，提出了两个可操作行为规则挖掘算法 MABR-1 和 MABR-2。最后，提出了可操作行为规则挖掘算法（模型）的经验验证方法，并验证了 MABRs 的有效性。该验证方法填补了可操作规则挖掘领域的空白。

⑥ 进一步建立了精确的可操作行为规则挖掘的计算模型，设计了多种有效、高效的规则挖掘算法。具体来说，为消解规则的冲突，提出了一种新的规则排序方法；为精确描述行为样本对可操作行为规则的非一致支持强度，提出了一个规则支持度的样本加权模型，以及一个相应的挖掘算法；为直接处理数值行为属性，提出了直接基于数值行为属性的可操作行为规则挖掘的新定义，以及一个相应的规则挖掘算法；为充分利用先验知识及显著减少挖掘算法的时间复杂度，提出了一个基于贝叶斯网络的可操作行为规则挖掘方法及相应算法；为显著减少挖掘算法的时间复杂度，提出了一个基于决策树的近似规则挖掘算法。

⑦ 探讨了大数据背景下的组织行为模式挖掘。具体讨论了组织行为模式挖掘面临的挑战、应对策略及实现方案等。

附录　MAROB 数据集中的相关属性表

类型	码	属性	含义	值	含义
行为属性	b_1	DOMORG-VIOLENCE	组织的国内暴力使用程度	0	未使用
				1	偶尔使用但不明确针对个人 （针对基础设施而且（或者）在攻击前发出警告）
				2	有规律地使用且针对安全人员（包括国家安全人员和非国家武装力量人员），但不包括政府非安全人员和平民
				3	有规律地使用且针对安全人员（包括国家安全人员和非国家武装力量人员）和（或）政府非安全人员，但不包括平民
				4	偶尔针对平民，主要针对安全人员或政府非安全人员
				5	有规律地针对平民
	b_2	TRANS-VIOLTARG	组织对跨国实体的暴力使用程度	0	未使用
				1	偶尔使用但不明确针对个人 （针对基础设施而且（或者）在攻击前发出警告）
				2	有规律地使用且针对安全人员（包括国家安全人员和非国家武装力量人员），但不包括政府非安全人员和平民
				3	有规律地使用且针对安全人员（包括国家安全人员和非国家武装力量人员）和（或）政府非安全人员，但不包括平民
				4	偶尔针对平民，主要针对安全人员或政府非安全人员
				5	有规律地针对平民
	b_3	TRANS-VIOLOC	组织的境外暴力使用程度	0	未使用
				1	偶尔使用但不明确针对个人（针对基础设施而且（或者）在攻击前发出警告）

（续表）

类型	码	属性	含义	值	含义
行为属性	b_3	TRANS-VIOLOC	组织的境外暴力使用程度	2	有规律地使用且针对安全人员(包括国家安全人员和非国家武装力量人员)，但不包括政府非安全人员和平民
				3	有规律地使用且针对安全人员(包括国家安全人员和非国家武装力量人员）和（或）政府非安全人员，但不包括平民
				4	偶尔针对平民，主要针对安全人员或政府非安全人员
				5	有规律地针对平民
Environment	e_1	ORST-POLSUP	外国政府是否提供政治支持	0	否
				1	是
	e_2	DIAFIN-SUP	移民社群是否提供非军事财政支持	0	否
				1	是
	e_3	ORG-CULTGR	组织的主流文化不满程度	0	未表达
				1	文化不满主要集中在消除歧视
				2	文化不满主要集中在建立或加强经济救济政策(如建立或增加国家文化保护或促进基金)

参考文献

[1] Schuler D. Social computing[J]. Comm ACM，1994，37: 28-29.

[2] Dryer D. C，Eisbach C.，Ark W. S. At what cost pervasive? A social computing view of mobile computing systems[J]. IBM Syst J，1999，38: 652-676.

[3] 王飞跃，李晓晨，毛文吉，等. 社会计算的基本方法与应用[M]. 浙江：浙江大学出版社，2013.

[4] Behrens T. E. J.，Hunt L. T.，Rushworth M. F. S. the computation of social behavior[J]. science，2009，324 (5931): 1160-1164.

[5] Vespignani A. Predicting the behavior of techno-social systems[J]. Science，2009，325(5939): 425-428.

[6] Mucha P. J.，Richardson T.，Macon K.，et al. Community structure in time-dependent，multiscale，and multiplex networks[J]. Science，328.

[7] Centola D. The spread of behavior in an online social network experiment[J]. Science，2010，329(5996): 1194-1197.

[8] Wang F.-Y.，Sun N.，Mao W.，et al. Editorial:special section on international partnership program[J]. Journal of Computer Science and Technology，2009，24(6): 997-999.

[9] Yang Q.，Zhou Z.，Mao W.，et al. Social learning[J].IEEE Intelligent Systems，2010，25(4): 9-11.

[10] Zeng D.，Chen H.，Lusch P.，et al. Social media analytics and intelligence[J]. IEEE Intelligent Systems，2010，25(6): 13-16.

[11] 王飞跃. 社会计算：科学·技术·人文[J]. 中国科学院院刊，2005，20(5): 370-375.

[12] 王飞跃. 社会计算的意义及其展望[J]. 中国计算机学会通讯，2006，2(2): 28-38.

[13] 王飞跃. 计算社会心理学的基本思想与方法[J]. 复杂性与智能化，2006，2: 7-9.

[14] 毛文吉.基于 MASIM 的社会推理与计算系统[J].系统科学与数学，2008，28(11): 1432-1440.

[15] Martinez V.，Simari G. I.，Sliva A.，et al. CONVEX: context vectors as a paradigm for learning organization behaviors based on similarity[J]. IEEE Intelligent Systems，2007，23(4): 51-57.

[16] Schrodt P. Forecasting conflict in the balkans using hidden markov models[D]. Programming for Peace: Computer-Aided Methods for International Conflict Resolution and Prevention[M]，Springer，2000: 161-184.

[17] Bond J.，Petroff V.，et al. Forecasting turmoil in indonesia: an application of hidden markov models[M]. Proc. Int'l Studies Assoc，2004: 17-21.

[18] Khuller S.，Martinez V.，Nau D.，et al. Finding most probable worlds of logic programs[M]. Proc. the First International Conference on Scalable Uncertainty Management，2007，45-59.

[19] Minorities at Risk Project. College Park，MD: Center for International Development and Conflict Management，2005. Retrieved from http://www.cidcm.umd.edu/mar/.

[20] Zadrozny B.，Elkan C. Learning and making decisions when costs and probabilities are both unknown in Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[M]，2001，204-213.

[21] Zadrozny B.，Langford J.，Abe N. A simple method for cost-sensitive learning[M]. Technical Report: IBM，2002.

[22] Abe N.，Zadrozny B.，Langford J. An iterative method for multi-class cost-sensitive learning[M]. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，2004，3-11.

[23] Zadrozny B.，Langford J.，Abe N. Cost-sensitive learning by cost—proportionate example weighting[C]. Proceedings of the 2003 IEEE International Conference on Data Mining (ICDM'03)，2003.

[24] Zadrozny B. One-benefit learning: cost-sensitive learning with restricted cost information[C].

Proceedings of the 1st International Workshop on Utility-based Data Mining，2005，53-58.

[25] Langford J.，Beygelzimer A. Sensitive error correcting output codes[C]. Proceedings of the 18th Annual Conference on Learning Theory，2005.

[26] Breiman L.，Friedman J. H.，Olshen R. A.，et al. Classification and Regression Trees[D]. Wadsworth，1983.

[27] Domingos P. MetaCost: a general method for making classifiers cost-sensitive[C]. Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，1999，155-164.

[28] Ting K. M. An instance-weighting method to induce cost-sensitive trees[J]. IEEE Transactions on Knowledge and Data Engineering，2002，14(3): 659-665.

[29] Elkan C. The foundations of cost-sensitive learning[C]. Proceedings of the 17th International Joint Conference on Artificial Intelligence，2001，973-978.

[30] Zhou Z.-H.，Liu X.-Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem[J]. IEEE Transactions on Knowledge and Data Engineering，2006，18(1):63-77.

[31] Drummond C.，Holte R. C. C4.5，class imbalance，and cost sensitivity: why under-sampling beats over-sampling[C]. Proceedings of Working Notes of the ICML'03 Workshop Learning from Imbalanced Data Sets，2003.

[32] Maloof M. A. Learning when data sets are imbalanced and when costs are unequal and unknown[C]. Proceedings of Working Notes ICML'03 Workshop Learning from Imbalanced Data Sets，2003.

[33] Chawla N.，Bowyer K.，Hall L.，et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research，2002，16: 321-357.

[34] Kaur H.，Actionable rules: issues and new directions[C]. Proceedings of World Academy of Science，Engineering and Technology，2005，61-64.

[35] Silberschatz A.，Tuzhilin A. What makes patterns interesting in knowledge discovery systems[J]. IEEE Transactions on Knowledge and Data Engineering，1996，8(6): 970-974.

[36] Ras Z.，Wieczorkowska A. Action-rules: How to increase profit of a company[M]. Principles

of Data Mining and Knowledge Discovery，Zighed D.，Komorowski J.，and Zytkow J.，Eds.: Springer，2000，75-116.

[37] Yang Q.，Cheng H. Mining case bases for action recommendation[C]. Proceedings of the Second IEEE International Conference on Data Mining (ICDM 02)，2002，522-529.

[38] Ling C.，Chen T.，Yang Q.，et al. Mining optimal actions for intelligent CRM[C]. Proceedings of the Second IEEE International Conference on Data Mining (ICDM 02)，2002，767-770.

[39] Liu B.，Hsu W.，Chen S. Using general impressions to analyze discovered classification rules[C]. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD 97)，1997，31-36.

[40] Ras Z.，Tsay L. Discovering extended action-rules (System DEAR)[C]. Proceedings of the International Intelligent Information Processing and Web Mining (IIPWM 03)，2003，293-300.

[41] Ras Z.，Dardzinska A. Action rules discovery，a new simplified strategy[J]. Foundations of Intelligent Systems，2006，4203: 445-453.

[42] Tzacheva A.，Ras Z.，"Constraint based action rule discovery with single classification rules[J].Rough Sets，Fuzzy Sets，Data Mining and Granular Computing.2007，4482: Springer，322-329.

[43] Ras Z.，Wyrzykowska E.，Wasyluk H. ARAS: action rules discovery based on agglomerative strategy[C]. Proceedings of the Third ECML/PKDD international conference on Mining complex data Warsaw，Poland: Springer-Verlag，2008.

[44] Im S.，Ras Z.，Wasyluk. H. Action rule discovery from incomplete data[M]. Knowledge and Information Systems，2009.

[45] Ras Z.，Tsay L. Mining e-action rules，system DEAR[J]. Data Mining: Foundations and Practice. 2008，118: 289-298.

[46] Tzacheva A.，Ras Z. Action rule mining[J]. International Journal of Intelligent Systems，2005，20(7):719-736.

[47] Ras Z.，Tzacheva A.，Tsay L.，et al. Mining for interesting action rules[C]. Proceedings of

IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2005)，2005，187-193.

[48] Ras Z.，Tzacheva A. In search for action rules of the lowest cost[J]. Monitoring，Security，and Rescue Techniques in Multiagent Systems. 2005，28: Springer，261-272.

[49] Tzacheva A.，Tsay L.-S. Tree-based construction of low-cost action rules[J]. Fundamenta Informaticae，2008，86(1，2): 213-225.

[50] He Z.，Xu X.，Deng S.，et al. Mining action rules from scratch[J]. Expert Systems with Applications，2005，29(3): 691-699 .

[51] Ras Z.，Dardzinska A.，Tsay L.-S.，et al. Association action rules[J]. Proceedings of IEEE/ICDM Workshop on Mining Complex Data (MCD 08) Pisa，Italy，2008，283-290.

[52] Ras Z.，Dardzinska A. Action rules discovery without pre-existing classification rules[J]. Proceedings of RSCTC 2008 Conference Akron，Ohio，2008，181-190.

[53] Yang Q.，Yin J.，Ling C.，et al. Postprocessing decision trees to extract actionable knowledge[J]. Proceedings of the Third IEEE International Conference on Data Mining (ICDM 03)，2003，685-688.

[54] Cao L.，ZhaoY.，Zhang H.，et al. Flexible frameworks for actionable knowledge discovery[J]. IEEE Transactions on Knowledge and Data Engineering，2010，22(9): 1299-1312.

[55] Liu B.，Hsu W.，Ma Y. Integrating classification and association rule mining[C]. Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD 98)，1998，80-86.

[56] Li W.，Han J.，Pei J. CMAR: Accurate and efficient classification based on multiple class-association rules[C]. Proceedings of the First IEEE International Conference on Data Mining (ICDM 01)，2001，369-376.

[57] Li J.，Topor R.，Shen H. Construct robust rule sets for classification[C]. Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining Edmonton，Alberta，Canada: ACM，2002，564-569.

[58] Wang K.，Zhou S.，He Y. Growing decision trees on support-less association rules[C].

Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining Boston，Massachusetts，United States: ACM，2000，265-269.

[59] Antonie M.-L.，Zaïane O. R. An associative classifier based on positive and negative rules[C]. Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery Paris，France: ACM，2004，64-69.

[60] Baralis E.，Garza P. A lazy approach to pruning classification rules[C]. Proceedings of the Second IEEE International Conference on Data Mining (ICDM 02)，2002，35.

[61] Baralis E.，Chiusano S.，Garza P. On support thresholds in associative classification[C]. Proceedings of the 2004 ACM Symposium on Applied Computing Nicosia，Cyprus: ACM，2004，553-558.

[62] 王珏. 机器学习机器应用[M]. 清华大学出版社. 2006.

[63] Duda R. O.，Hart P. E.，Stork D. G. Pattern Classification (Second Edition)[M]. Wiley，2001.

[64] Webb A. Statistical Pattern Recognition (Second Edition)[M]. Wiley，2002.

[65] Jain A. K.，Duin R. P. W.，Mao J. Statistical pattern recognition: a review[J]. IEEE Trans. PAMI，2000，22(1): 4-37.

[66] Bishop C.M. Pattern Recognition and Machine Learning[M]. Springer，2006.

[67] Hastie T.，bshrani T.，Friedman J. The Elements of Statistical Learning[M]. Springer，2001.

[68] Fukunaga K. Introduction to Statistical Pattern Recognition[M]. Academic Press，1990.

[69] Theodoridis S. Pattern Recognition (Second Edition)[M]. Academic Press，2003.

[70] Vapnik V. N. Statistical Learning Theory[M]. Wiley: New York，1998.

[71] Plewczynski D.，Tkacz A.，Godzik A.，et al. A support vector machine approach to the identification of phosphorylation sites[J]. Cell. Mol. Biol. Lett.，2005，10: 73-89.

[72] Chang C. C.，Lin C. J. Training nu-Support vector classifiers: Theory and Algorithms[J]. Neural Computation，2001，13: 2119-2147.

[73] Guha R.，Jurs P. C. Interpreting computational neural network QSAR models: A measure of descriptor importance[J]. J. Chem. Inf. Model.，2005，45: 800-806.

[74] Kauffman G. W.，Jurs P. C. QSAR and k-Nearest neighbor classification analysis of selective

cyclooxygenase-2 inhibitors using topologically based numerical descriptors[J]. J. Chem. Inf. Comput. Sci.，2001，41: 1553-1560.

[75] Agrafiotis D. K.，Cedeno W.，Lobanov V. S. On the use of neural network ensembles in QSAR and QSPR[J]. J. Chem. Inf. Comput. Sci.，2002，42: 903-911.

[76] Schneider G.，Wrede P. Artificial neural networks for computer-based molecular design[J]. Prog. Biophys. Mol. Biol.，1998，70: 175-222.

[77] Haykin S. Neural Networks，A Comprehensive Foundation[M]. (2nd eds.)，New York: Printice Hall，1999.

[78] Sheridan R. P.，Feuston B. P.，Maiorov V. N.，et al. Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR[J]. J. Chem. Inf. Comput. Sci.，2004，44: 1912-1928.

[79] Wang D. L. Pattern recognition: neural networks in perspective[M]. IEEE Expert，1993，52-60.

[80] Raymer M. L.，Sanschagrin P. C.，Punch W. F.，et al. Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest-neighbors genetic algorithm[J]. J. Mol. Biol.，1997，265: 445-464.

[81] Labute P. Binary QSAR: a new method for the determination of quantitative structure activity relationships[C]. Proc.Pac. Symp. Biocomput.，1997，444-455.

[82] Sheridan R. P.，Nachbar R. B.，Bush B. L. Extending the trend vector: the trend matrix and sample-based partial least squares[J]. J. Comput.-Aided Mol. Des.，1994，8: 323-340.

[83] Rusinko A.，Farmen M. W.，Lambert C. G.，et al. Analysis of a large structure/biological activity data set using recursive partitioning[J]. J. Chem. Inf. Comput. Sci.，1999，39: 1017-1026.

[84] Quinlan J. R. C4.5: Programs for Machine Learning[M]. Morgan Kaufmann Publishers Inc.，1993.

[85] Svetnik V.，Liaw A.，Tong C.，et al. Random forest: a classification and regression tool for compound classification and QSAR modeling[J]. J. Chem. Inf. Comput. Sci.，2003，43: 1947-1958，.

[86] Breiman L. Random forests[J]. Machine Learning，2001，45: 5-32.

[87] Hand D. J.，Yu K. Idiot’s bayes - not so stupid after all?[J]. International Statistical Review，2001，69(3): 385-398.

[88] Vapnik V. N. The nature of statistical learning theory[M]. Springer-Verlag，1998，1-187.

[89] Jain A. K.，Mao J.，Mohiuddin K. M. Artificial neural networks: a tutorial[J]. Computer，1996，29(3):31-44.

[90] Christopher M. B. Neural networks for pattern recognition[M]. Oxford University Press，1995，1-482.

[91] Kotsiantis S.，Zaharakis I.，Pintelas P. Machine learning: a review of classification and combining techniques[J]. Artificial Intelligence Review，2006，26(3): 159-190.

[92] Thabtah F. A review of associative classification mining[J]. The Knowledge Engineering Review，2007，22(1): 37-65.

[93] Ling C. X.，Huang J.，Zhang H. AUC: a statistically consistent and more discriminating measure than accuracy[C]. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence，Acapulco，Mexico，August 9-15，2003，329-341.

[94] Japkowicz N.，Stephen S. The class imbalance problem: a systematic study[J]. Intelligent Data Analysis，2002，6(5): 429-450.

[95] Kohavi R.，Wolpert D. H. Bias plus variance decomposition for zero-one loss functions[C]. Proceedings of the Thirteenth International Conference on Machine Learning，Bari，Italy，July 3-6，1996，275-283.

[96] Govindarajan M. Text mining technique for data mining application[C]. Proceedings of World Academy of Science，Engineering and Technology，2007，26(104): 544-549.

[97] Liu B.，Hsu W.，Ma Y. Integrating classification and association rule mining[C]. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining，New York，USA，August 27-31，1998，80-86.

[98] Li W.，Han J.，Pei J. CMAR: accurate and efficient classification based on multiple class-association rules[C]. Proceedings of the First IEEE International Conference on Data Mining，California，USA，November 29-December 2，2001，369-376.

[99] Yin X.，Han J. CPAR: classification based on predictive association rules[C]. Proceedings of the Third SIAM International Conference on Data Mining，San Francisco，USA，May 1-3，2003，369-376.

[100] Grčar M.，Mladenič D.，Fortuna B.，et al. Data sparsity issues in the collaborative filtering framework[J]. Proceedings of the Seventh International Workshop on Knowledge Discovery on the Web，Chicago，USA，August 21，2005，58-76.

[101] Nathalie J. Class imbalances: are we focusing on the right issue?[C]. Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets，Washington DC，USA，August 21，2003.

[102] Japkowicz N. Class imbalance problem: significance and strategies[C]. Proc. the Second International Conference on Artificial Intelligence，Las Vegas，USA，June 26-29，2000，111-117.

[103] Zhang J. kNN approach to unbalanced data distributions: a case study involving information extraction[C]. Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets，Washington DC，USA，August 21，2003.

[104] Wolpert D. Stacked Generalization[J]. Neural Networks，1992，5(2): 241-260.

[105] Sarker R. A.，Abbass H.A.，Newton C. S.，et al. Heuristics and optimization for knowledge discovery[M]. Idea Organization Inc (IGI)，2002.

[106] Liu X-Y.，Wu J-X.，Zhou Z-X. Exploratory under-sampling for class-imbalance learning[C]. Proceedings of the Sixth IEEE International Conference on Data Mining，Hong Kong，China，2006.

[107] Provost F.，Fawcett T.，Kohavi R. The case against accuracy estimation for comparing induction algorithms[C]. Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann，San Francisco，1998，445-453.

[108] Weiss G. M. Mining with rarity – problems and solutions: A unifying framework[J]. SIGKDD Explorations，2004，6(1): 7-19.

[109] Drummond C.，Holte R. C. Explicitly representing expected cost: an alternative to roc representation[C]. Proceedings of the ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining，2000，198-207.

[110] Provost F.，Fawcett T. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions[C]. Proceedings of the Third ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining，1997，43-48.

[111] Pawlak Z. Information systems theoretical foundations[J]. Information Systems，1981，6(3): 205-218.

[112] Agrawal R.，Srikant R. Fast algorithms for mining association rules[C]. Proceeding of the Twentieth International Conference on VLDB，1994，487-499.

[113] Agrawal R.，Imielinski T.，Swami A. Mining association rules between sets of items in large databases[J]. ACM SIGMOD Record，1993，22(2): 207-216.

[114] Han J.，Pei J.，Yin Y.，et al. Mining frequent patterns without candidate generation: A frequent-pattern tree approach[J]. Data Mining and Knowledge Discovery，2004，8: 53-87.

致谢

本书得到国家自然科学基金项目（编号：71462001）的资助。